

**UNIVERSIDAD POLITÉCNICA SALESIANA
SEDE QUITO**

**CARRERA:
INGENIERÍA DE SISTEMAS**

**Trabajo de titulación previo a la obtención del título de:
INGENIEROS DE SISTEMAS**

**TEMA:
DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA INFORMÁTICO PARA EL
MANEJO DE INVENTARIOS DE LA DISTRIBUIDORA “MATEO”**

**AUTORES:
JONATHAN DANIEL ARGOTI CAIZA
JOSÉ GEOVANNI PORTILLA ROMÁN**

**TUTORA:
LINA PATRICIA ZAPATA MOLINA**

Quito, abril del 2018

CESIÓN DE DERECHOS DE AUTOR

Nosotros: Jonathan Daniel Argoti Caiza, con documento de identificación N° 1726482092 y José Geovanni Portilla Román, con documento de identificación N° 1724468366, manifestamos que voluntariamente otorgamos a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que somos autores del trabajo de titulación con el tema: “DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA INFORMÁTICO PARA EL MANEJO DE INVENTARIOS DE LA DISTRIBUIDORA “MATEO”, mismo que ha sido desarrollado para optar por el título de INGENIEROS DE SISTEMAS en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En aplicación a lo determinado en la Ley de Propiedad Intelectual, en nuestra condición de autores nos reservamos los derechos morales de la obra antes citada.

En concordancia, suscribimos este documento en el momento que hacemos la entrega del trabajo final en formato impreso y digital a la Biblioteca de la Universidad Politécnica Salesiana.



JONATHAN DANIEL
ARGOTI CAIZA
CI: 1726482092



JOSÉ GEOVANNI
PORTILLA ROMÁN
CI: 1724468366

Quito, abril del 2018

DECLARATORIA DE COAUTORIA DE LA TUTORA

Yo, Lina Patricia Zapata Molina, declaro que bajo mi dirección y asesoría fue desarrollado el Proyecto Técnico, con el tema: “DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA INFORMÁTICO PARA EL MANEJO DE INVENTARIOS DE LA DISTRIBUIDORA “MATEO”, realizado por: Jonathan Daniel Argoti Caiza y José Geovanny Portilla Román, obteniendo un producto que cumple con todos los requisitos estipulados por la Universidad Politécnica Salesiana, para ser considerados como trabajo final de titulación.

Quito, abril del 2018



LINA PATRICIA ZAPATA MOLINA

CI: 0501877278

DEDICATORIA

Dedico este trabajo a Dios por haberme guiado por el camino correcto para lograr culminar mi carrera, a mis padres por haber sido mi apoyo durante toda esta etapa y haberme aconsejado cuando lo necesitaba, ellos fueron quienes me inculcaron valores y disciplina que fueron necesarios aplicar en diversas situaciones. A mi familia por confiar siempre en mí. A ti Asthin que desde que te conocí te convertiste en una motivación y me impulsaste a ser mejor persona.

Jonathan Daniel Argoti Caiza

Dedico este trabajo a mis Padres, por haberme brindado todo su apoyo incondicional, por acompañarme durante todo mi trayecto estudiantil y de vida. A mi Madre quien con su sabiduría supo ayudarme en esos momentos más difíciles además de ser el pilar fundamental para comenzar y culminar mi carrera profesional.

José Geovanni Portilla Román.

AGRADECIMIENTO

Agradezco a la Universidad Politécnica Salesiana por las enseñanzas académicas y morales que me inculco, por todo el tiempo que estuve en sus aulas preparándome para la vida profesional, a la Ing. Lina Zapata por haberme dado la confianza para realizar este trabajo y por haberme guiado para el desarrollo del mismo.

Jonathan Daniel Argoti Caiza

José Geovanni Portilla Román

ÍNDICE

Introducción	1
Objetivo General	2
Objetivos Específicos.....	2
1 Capítulo 1 Marco Conceptual	3
1.1 Manejo de inventarios	3
1.2 Metodología de desarrollo en cascada.....	3
1.2.1 Análisis.....	4
1.2.2 Diseño.	5
1.2.3 Metodología de diseño OOHDM.	5
1.2.3.1 Implementación.	7
1.2.3.1.1 <i>Herramientas de desarrollo.</i>	7
1.2.3.2 Pruebas.	9
2 Capítulo 2 Análisis y Diseño	10
2.1 Análisis	10
2.1.1 Personas involucradas en el proyecto.	10
2.1.2 Cronograma de actividades.	11
2.1.3 Especificación de requerimientos.	13
2.1.4 Segmentación del sistema.	24
2.2 Diseño.....	26
2.2.1 Diagramas de casos de uso.....	26
2.2.2 Diagramas de secuencia.	38
2.2.3 Diseño conceptual.	42
2.2.4 Diseño navegacional.	44
2.2.5 Diseño de interfaces abstracta.	45
3 Capítulo 3 Construcción del Sistema.....	49
3.1 Configuración de herramientas de desarrollo	49
3.2 Código del sistema	51
3.3 Pruebas de software.....	57
CONCLUSIONES	59
RECOMENDACIONES	60
LISTA DE REFERENCIAS	61
ANEXOS..	62

ÍNDICE DE TABLAS

Tabla 1. Descripción de pruebas	9
Tabla 2. Matriz de involucrados.....	10
Tabla 3. Detalle de actividades	11
Tabla 4. Descripción del requerimiento de autenticación de usuarios	13
Tabla 5. Descripción del requerimiento para ingresar productos	14
Tabla 6. Descripción del requerimiento para consultar productos.....	14
Tabla 7. Descripción del requerimiento para actualizar productos	15
Tabla 8. Descripción del requerimiento para eliminar productos	15
Tabla 9. Descripción del requerimiento para ingresar clientes	16
Tabla 10. Descripción del requerimiento para consultar clientes	16
Tabla 11. Descripción del requerimiento para actualizar clientes	17
Tabla 12. Descripción del requerimiento para eliminar clientes.....	17
Tabla 13. Descripción del requerimiento para ingresar proveedores.....	18
Tabla 14. Descripción del requerimiento para consultar proveedores	18
Tabla 15. Descripción del requerimiento para actualizar proveedores	19
Tabla 16. Descripción del requerimiento para eliminar proveedores	20
Tabla 17. Descripción del requerimiento de escalabilidad del sistema	20
Tabla 18. Descripción del requerimiento de facilidad de uso del sistema	21
Tabla 19. Descripción del requerimiento de seguridad del sistema.....	21
Tabla 20. Descripción del requerimiento de validación del sistema.....	22
Tabla 21. Descripción del requerimiento de disponibilidad del sistema	22
Tabla 22. Descripción del requerimiento de eficiencia del sistema.....	23
Tabla 23. Descripción del requerimiento de eficacia del sistema	23
Tabla 24. Descripción del requerimiento de integridad del sistema	24
Tabla 25. Detalle de pruebas de caja negra.....	57
Tabla 26. Detalle de pruebas de caja blanca	57
Tabla 27. Detalles de pruebas de integración.....	58

ÍNDICE DE FIGURAS

Figura 1. Metodología en cascada.....	3
Figura 2. Gestión del sistema.....	24
Figura 3. Ingreso de usuario.....	27
Figura 4. Gestionar productos.....	27
Figura 5. Ingresar productos.....	28
Figura 6. Consultar productos.....	29
Figura 7. Actualizar productos.....	29
Figura 8. Eliminar productos.....	30
Figura 9. Registro De Salida.....	31
Figura 10. Gestionar clientes.....	31
Figura 11. Ingresar cliente.....	32
Figura 12. Consultar clientes.....	33
Figura 13. Actualizar clientes.....	33
Figura 14. Eliminar clientes.....	34
Figura 15. Gestionar proveedores.....	35
Figura 16. Ingresar proveedores.....	35
Figura 17. Consultar proveedores.....	36
Figura 18. Actualizar proveedores.....	37
Figura 19. Eliminar proveedores.....	37
Figura 20. Conceder permisos a usuarios.....	38
Figura 21. Acceder al sistema.....	39
Figura 22. Insertar datos.....	39
Figura 23. Consultar información.....	40
Figura 24. Actualizar información.....	41
Figura 25. Eliminar información.....	41
Figura 26. Diagrama BDD.....	43
Figura 27. Diagrama de clases.....	44
Figura 28. Diagrama navegacional.....	45
Figura 29. Interfaz ingreso.....	46
Figura 30. Interfaz menú administrador.....	46
Figura 31. Interfaz menú vendedor.....	47
Figura 32. Interfaz productos.....	48
Figura 33. Interfaz NetBeans.....	49
Figura 34. Interfaz xampp.....	50
Figura 35. Interfaz mysql.....	50
Figura 36. Código bdd.....	51
Figura 37. Código login.....	52
Figura 38. Código acceso.....	53
Figura 39. Código insertar.....	54
Figura 40. Código actualizar.....	54
Figura 41. Código eliminar.....	55

Figura 42. Código consultar56

Figura 43. Código salida56

Resumen

El presente documento contiene toda la información relacionada a la creación de un sistema para el manejo de inventarios. Se muestra los antecedentes del problema del control de inventarios y se justifica la solución planteada, además se detalla la creación del sistema para el cual se usó la metodología de desarrollo en cascada. Con la ayuda de diagramas de base de datos, de casos de uso, de secuencia y prototipos de interfaz se diseñó la solución. Posteriormente se menciona como se integraron cada uno de los componentes mencionados para dar como resultado el sistema.

Este proyecto se enfoca en el desarrollo de un sistema de inventarios para la distribuidora “Mateo”. El manejo de inventarios se ha llevado de forma manual, lo cual funcionaba correctamente hasta que los resultados que se obtenían no eran los mejores. Debido al crecimiento de ventas de la distribuidora y luego de realizar un estudio se ha visto la necesidad de automatizar el proceso de manejo de inventarios.

Esto se hará en busca de optimizar este proceso, lo que dará como resultado tener un adecuado control de los productos que tiene la distribuidora, manteniendo el stock actualizado e información oportuna cuando se la requiera utilizar. De esta manera el gerente tendrá una perspectiva más amplia para tomar decisiones que sean beneficiosas para la distribuidora.

Abstract

This document contains all the information related to the creation of a system for the management of inventories. The background of the problem of inventory control is shown and the proposed solution is justified, and the creation of the system for which the cascade development methodology was used is detailed. With the help of database diagrams, use cases, sequence and interface prototypes, the solution was designed. Subsequently, it is mentioned how each of the aforementioned components were integrated to result in the system.

This project focuses on the development of an inventory system for the distributor "Mateo". Inventory management has been carried out manually, which worked correctly until the results obtained were not the best. Due to the sales growth of the distributor and after conducting a study has seen the need to automate the process of inventory management.

This will be done in order to optimize this process, which will result in having an adequate control of the products that the distributor has, keeping the stock updated and timely information when required to use. In this way the manager will have a broader perspective to make decisions that are beneficial to the distributor.

Introducción

Inicialmente el manejo de inventarios apareció bajo la necesidad de tener un control de los productos que tiene un negocio. Para hacer un inventario se requería de una gran cantidad de tiempo. Este proceso se ha llevado de forma manual durante mucho tiempo, lo que se hacía era registrar los productos con sus respectivas características en un listado que servía de referencia para conocer el stock de cada producto.

Los resultados de un inventario manual no eran totalmente confiables, las personas que se encargaban de hacer el inventario podían cometer errores. Cuando el gerente del negocio requería un reporte de las compras y ventas de los productos, el gerente tenía que solicitarlo a los empleados, los cuales tenían que pasar horas trabajando para generar ese reporte. Lo cual no era óptimo para el negocio.

Debido al crecimiento de las ventas en la distribuidora “Mateo” y la falta de manejo de inventarios ha ocasionado que existan perdidas de los productos y que la propietaria desconozca las existencias reales en bodega de sus productos.

Lo que se hace actualmente es llevar un registro de los productos que ingresan a bodega en un cuaderno. Cuando la distribuidora realiza un pedido de mercadería el vendedor la recibe y la registra, de igual manera cuando se hace una venta se registra la salida de la mercadería. No se tiene información organizada a la que se pueda acceder cuando sea necesario.

El realizar este proceso manual consume mucho tiempo, el cual podría ser utilizado en otra actividad que beneficie a la distribuidora. El gerente de la distribuidora se ha dado

cuenta de este problema, por lo que ha decidido buscar una solución y ha recurrido a implementar un sistema de información para el control de inventarios.

La automatización de este proceso es importante para mejorar la productividad, la atención a los clientes y para evitar pérdidas de mercadería. Esto permite que diariamente se lleve un registro de los productos, por lo que se tendrá un control de las entradas y salidas de cada producto. Además, se cuenta con un registro de clientes y proveedores a los cuales se puede acceder cuando sea necesario. También se reduce el tiempo de trabajo que los empleados requerían para hacer un inventario y ahora se dedican a otras actividades dentro de la distribuidora.

Objetivo General

Desarrollar un sistema informático para el control de inventarios mediante la utilización de herramientas de software libre, para administrar adecuadamente, evitando el aumento de costos o pérdidas en la mercadería.

Objetivos Específicos

Desarrollar un sistema informático utilizando herramientas de libre distribución que permitan ahorrar recursos.

Realizar un inventario manual de toda la mercadería existente en el almacén y clasificarla según su tipo en el sistema.

Optimizar el tiempo empleado en la entrada y salida de mercadería en el almacén para emplearlo en otras actividades.

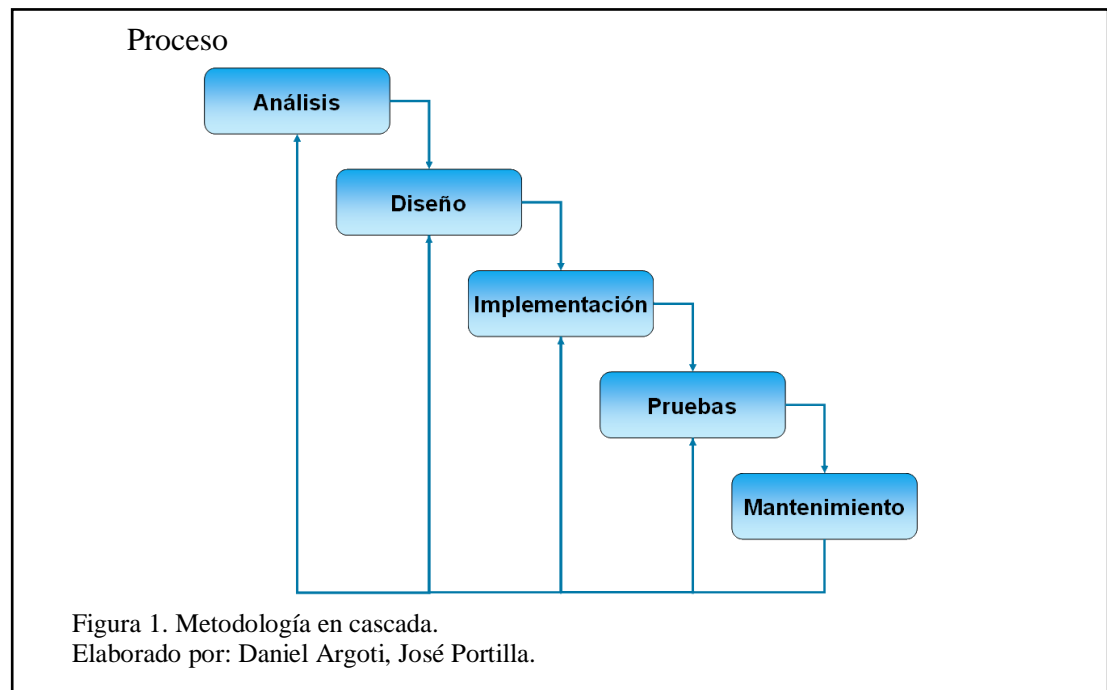
Capítulo 1

Marco Conceptual

1.1 Manejo de inventarios

La optimización de recursos es indispensable para mejorar la productividad, los recursos bien empleados evitarán pérdidas y aumentarán las ganancias. La automatización de procesos mejorará la atención que se brinda a los clientes y se tendrá un control automático sobre las actividades del negocio. Además, se reduce el trabajo de los empleados para que puedan dedicarse a otra actividad usando de mejor manera su tiempo. También se reduce la pérdida de mercadería y se tendrá productos siempre disponibles. Cuando se necesite adquirir nuevos productos se lo hará oportunamente. (Miguez & Bastos, 2006)

1.2 Metodología de desarrollo en cascada



En la figura No. 1 se muestra un modelo lineal-secuencial que representa el ciclo de vida del desarrollo de software, se basa en un desarrollo rígido que sigue una estructura básica compuesta por un conjunto de actividades: el análisis es la recolección de los

requerimientos sobre el producto, el diseño son diagramas que describen la estructura del software (se usara la metodología de diseño OOHDM), la implementación es la construcción del software, las pruebas son la evaluación sistemática del software y el mantenimiento es control periódico de su funcionamiento. (Goytia & Gutierrez, 2014)

1.2.1 Análisis.

Dentro del análisis se debe identificar a las personas involucradas en el proyecto, los requerimientos del software se obtendrán mediante reuniones con esas personas, además se debe distribuir el trabajo mediante un cronograma de actividades. Esto se debe hacer hasta obtener toda la información necesaria para iniciar con la etapa de diseño.

Identificación de involucrados

Permite clasificar a las personas o grupos según sus intereses, algunos serán beneficiados o perjudicados y tienen mucha influencia en el proyecto. Las acciones que ejecuten cada uno de los involucrados pueden hacer que el proyecto avance o se detenga. La mejor forma de que un proyecto tenga éxito, es identificando a los involucrados para llegar a un consenso con ellos que permita que el proyecto no sea afectado. (Miranda, 2005)

Especificación de requerimientos

Se debe identificar todos los requerimientos necesarios para el desarrollo del software. También debemos conocer a todos los actores y sus funciones a realizar. Para representar la interacción entre el usuario y el cliente, se usan diagramas de casos de uso donde se relaciona estas actividades. Es muy importante saber interpretar toda la información recopilada, porque de esto depende el éxito del proyecto. Si es necesario hay que volver a recopilar los requerimientos.

Existen dos tipos de requerimientos funcionales y no funcionales, los funcionales se enfocan en lo que el usuario necesita que el sistema haga, es una visión general de las funcionalidades del sistema, mientras que los requerimientos no funcionales se enfocan en los requerimientos que el desarrollador recomienda al cliente en base a su experiencia, es una visión más específica. Por supuesto que estos requerimientos deben ser aprobados conjuntamente. (Kendall, 2007)

Cronograma de actividades

Es una lista de actividades que el equipo de trabajo tiene que realizar, se hace un cronograma en el que a cada tarea se le asigna una fecha de iniciación y de finalización, además se asigna un responsable para esa tarea. Y el estado de la tarea para tener un seguimiento de cumplimiento y alcance. (Kendall, 2007)

1.2.2 Diseño.

En esta etapa del desarrollo del software se hizo uso de la metodología de diseño OOHDM, la cual permitirá diseñar la estructura del sistema mediante el uso de diseños y diagramas.

1.2.3 Metodología de diseño OOHDM.

Es muy importante para el diseño de aplicaciones, por lo que toda persona que se dedique al desarrollo web tiene que aplicar esta metodología. Dentro del modelo OOHDM existe un proceso con varias etapas que se debe ejecutar para realizar el diseño. Además, cada etapa exige ciertos requisitos que son necesarios para el cumplimiento de cada función, estas funciones representan a los requerimientos que tiene el usuario.

Este modelo fue diseñado por D. Schwabe, G. Rossi, and S. D. J. Barbosa, con el objetivo de ayudar en el diseño de aplicaciones hipermedia (páginas web). Estas aplicaciones son sistemas basados en brindar servicios a un equipo hombre-máquina. Estas aplicaciones logran integrar lo formal e informal generado por la máquina y el hombre respectivamente. (Lamarca, 2013)

Diagramas propuestos por OOHDM:

Diseño conceptual

Se crea un esquema conceptual (modelo entidad - relación) representado en clases, relaciones y subsistemas. No se debe priorizar en las actividades de los actores del sistema, lo que hay que hacer es priorizar en la estructura de datos de la aplicación. Este diseño sirve para tener una visión clara de la estructura del sistema. En base a estas entidades se obtendrán las vistas que nos permitirán interactuar con el sistema. (Lamarca, 2013)

Diseño navegacional

Se basa en dos esquemas el diagrama de clase y el diagrama de navegación, cada uno tiene nodos, enlaces y estructuras de acceso según el diseño conceptual, que permite navegar en la página, donde los enlaces son relaciones y los nodos son ventanas lógicas. Se realiza una estructura que servirá de base para navegar en el sistema. (Lamarca, 2013)

Diseño de interfaz abstracta

Se muestra cómo van a aparecer las ventanas finales del sistema, describe como debe ser la navegación dentro de la interfaz. Se representa el comportamiento de la interfaz, se usa ventanas, cuadros de texto, botones y otros componentes de diseño. Es el prototipo inicial que servirá de guía para el posterior desarrollo del sistema. Cada ventana que se crea vendrá

dada luego de un análisis que empezó en los requerimientos y siguió una metodología de diseño. (Lamarca, 2013)

Implementación

Se relaciona los objetos de interfaz con los objetos de implementación. Esta implementación de interfaces permite mostrar la interacción cliente – servidor. La interfaz comienza a realizar las actividades que se plantearon en los requerimientos. Se comienza a construir el software, toda la estructura generada a algo funcional, es decir implementando mediante código todo lo que debe hacer el sistema. (Lamarca, 2013)

1.2.3.1 Implementación.

Para la construcción del sistema se utilizará las siguientes herramientas que se detallan a continuación, basados en una programación de código abierto.

1.2.3.1.1 Herramientas de desarrollo.

NetBeans

Es un entorno de desarrollo integrado para el desarrollo de aplicaciones que se adaptan a cualquier tipo de tecnología. NetBeans tiene una arquitectura segura y flexible, esto facilita el desarrollo de aplicaciones mediante el uso de componentes de software. El modularidad es una de las principales virtudes de este entorno. (Oracle, NetBeans, 2017)

XAMPP

Es una distribución de software libre de Apache compuesta por varios complementos como: MariaDB, PHP y Perl (lenguajes de programación), es de gran ayuda en la gestión de base de datos web. Además, es un servidor web de plataforma de software libre que

funciona en la gran mayoría de sistemas operativos, permitiendo realizar una interacción cliente-servidor virtual. (Apache, 2017)

MySQL

Es un sistema de gestión de base de datos que se basa en un lenguaje de consulta basado en jerarquías, es muy usado especialmente para desarrollo web por ser ligero. Puede ser usado en varias plataformas, tiene un entorno de desarrollo fácil de usar, lo que facilita la creación y manipulación de una base de datos. Toda la estructura esta almacenada en un servidor propio. (Oracle, 2017)

Hojas de estilo en cascada (CSS)

Son archivos que sirven para describir cómo será la presentación de documentos estructurados. CSS es de gran utilidad para separar el fondo de la forma, es decir el contenido del diseño. Estos estilos pueden ser reutilizados en varios documentos estructurados. (Network, 2017)

JavaScript

Es un lenguaje de programación que permite crear aplicaciones que interactúan con el usuario automáticamente, esto se hace con la ayuda de algoritmos que se activan al realizar acciones específicas. Esto permite la interacción entre el usuario y la aplicación. Javascript no requiere ser compilado (ejecutar manualmente), se ejecuta directamente al iniciar el sistema. (School, 2017)

Bootstrap

Es un conjunto de herramientas de código libre para el diseño de aplicaciones, fue desarrollado por GitHub, tiene soporte para HTML5 y CSS3 que son las tecnologías más modernas. Está compuesto por plantillas de diseño para formularios, menús, tablas, botones. Se puede aplicar cualquiera de estas plantillas en todo tipo de aplicación. (GitHub, 2017)

AdminLTE

Es una platilla web de código libre muy usada para paneles de administración, desarrollada sobre HTML, con funcionalidades de CSS, JavaScript y Bootstrap. Tiene un diseño intuitivo para el usuario que le permitirá realizar una correcta gestión del sistema, además AdminLTE tiene módulos que permiten configurar su funcionalidad y adaptarlo a la lógica del negocio. (Almsaeed, 2017).

1.2.3.2 Pruebas.

Tabla 1. Descripción de pruebas

Pruebas	Descripción
Pruebas unitarias	Se enfoca en una unidad de código, se clasifica en pruebas de caja negra y caja blanca.
Pruebas de caja negra	Son pruebas que se aplican sin tomar en cuenta el código fuente, solo se enfocan en las entradas y salidas del sistema.
Pruebas de caja blanca	Son pruebas que hacen un seguimiento al código fuente para verificar el orden de instrucción de cada línea de código.
Pruebas de integración	Sirven para verificar el correcto funcionamiento de un conjunto de componentes dentro del sistema, los cuales deben trabajar juntos.

Nota: se define el enfoque de cada una de las pruebas.

En la tabla No. 1 se detallan los tipos de prueba que se implementaron en el desarrollo de este sistema, se realizaron pruebas como: pruebas de caja blanca, pruebas de caja negra y pruebas de integración.

Capítulo 2

Análisis y Diseño

2.1 Análisis

Se hace un estudio a profundidad de lo que requiere el cliente respecto a la automatización de los procesos, se toma en cuenta quienes están involucrados en el proyecto. Se establecen las actividades a realizarse dentro de un cronograma, además se clasifica los requerimientos según su prioridad y se diagrama las acciones que deberá realizar el sistema. Todos los datos que se obtienen previamente sirven para ser analizados y posteriormente utilizados en el diseño y construcción del sistema propuesto. (Cortes, 2006)

2.1.1 Personas involucradas en el proyecto.

Tabla 2. Matriz de involucrados

#	Involucrado	Interés	Impacto
1	Gerente de la Distribuidora Mateo.	Administrar el negocio eficientemente.	Aumentar ganancias y crecimiento del negocio.
2	Líder del proyecto.	Entregar un sistema bien desarrollado que cumpla con los objetivos planteados.	Satisfacer las necesidades del cliente.
3	Desarrolladores.	Construir un sistema de inventarios que tenga control de los productos de la distribuidora.	Cumplir con el plan de desarrollo del sistema.
4	Empleados.	Realizar las principales actividades dentro del almacén.	Mejorar la distribución de tiempo y optimización del trabajo.
5	Clientes.	Adquirir los productos al por mayor y menor.	Mejorar la atención y un servicio eficiente.

6	Proveedores.	Proveer productos al almacén oportunamente.	Stock de productos y actualización de pedidos oportuna.
---	--------------	---	---

Nota: Participantes del proyecto.

En la tabla No. 2 se identifica a las personas involucradas en el proyecto, organizadas en 6 grupos como: el líder del proyecto que dirige a los desarrolladores, el gerente que se encarga de administrar el negocio, los desarrolladores que son el equipo de trabajo, los empleados que trabajan en la distribuidora, los clientes que consumen productos y los proveedores que abastecen de productos a la distribuidora.

2.1.2 Cronograma de actividades.

Es el listado de actividades que el equipo de trabajo tiene que realizar. Y en base a esto se hace estimaciones para la ejecución de estas actividades, definiendo fechas de entrega. Además, se asigna un responsable a cada actividad y se hace controles sobre cada actividad para ver su cumplimiento.

Tabla 3. Detalle de actividades

#	Tarea	Fecha	Responsable	Estado
1	Reunión con el cliente para recolectar requerimientos	03/10/2016	Daniel Argoti - Giovanni Portilla	Terminado
2	Análisis de datos obtenidos del almacén	10/10/2016	Daniel Argoti	Terminado
3	Diseño de diagrama de casos de uso	17/10/2016	Daniel Argoti	Terminado
4	Diseño del modelo de base de datos	24/10/2016	Daniel Argoti	Terminado
5	Diseño del diagrama de clases	07/11/2016	Daniel Argoti Giovanni Portilla	Terminado
6	Diseño del modelo de interfaz	14/11/2016	Giovanni Portilla	Terminado
7	Reunión con el cliente para revisión	14/11/2016	Daniel Argoti	Terminado
8	Diseño de interfaz de navegación	21/11/2016	Giovanni Portilla	Terminado

9	Desarrollo modulo productos	05/12/2016	Daniel Argoti	Terminado
10	Reunión con el cliente para revisión	26/12/2016	Daniel Argoti	Terminado
11	Desarrollo modulo clientes	03/01/2017	Giovanni Portilla	Terminado
11	Reunión con el cliente para revisión	24/01/2017	Giovanni Portilla	Terminado
12	Desarrollo modulo proveedores	31/01/2017	Giovanni Portilla	Terminado
13	Reunión con el cliente para revisión	21/02/2017	Giovanni Portilla	Terminado
14	Prueba unitaria	28/02/2017	Daniel Argoti	Terminado
15	Integración de módulos	21/03/2017	Giovanni Portilla	Terminado
16	Pruebas de integración	11/04/2017	Daniel Argoti	Terminado
17	Inventario del almacén	25/04/2017	Daniel Argoti - Giovanni Portilla	Terminado
18	Cargar la información obtenida en el inventario al sistema	09/05/2017	Giovanni Portilla	Terminado
19	Pruebas del sistema	16/05/2017	Daniel Argoti	Terminado
20	Revisión final con el cliente	30/05/2017	Daniel Argoti - Giovanni Portilla	Terminado
22	Capacitación a los usuarios	05/06/2017	Daniel Argoti - Giovanni Portilla	Terminado
22	Entrega del sistema de inventarios	12/06/2017	Daniel Argoti	Terminado
23	Implementación del sistema	19/06/2017	Daniel Argoti - Giovanni Portilla	Terminado

Nota: Información del cronograma de actividades.

La tabla No. 3 muestra un listado de actividades que se realizaron en el desarrollo del sistema. Los encargados de realizar estas actividades fueron: Daniel Argoti y Geovanni Portilla. Y la persona encargada de revisar si se han cumplido o no con estas actividades deberá llevar un control periódico.

2.1.3 Especificación de requerimientos.

Es el listado de requerimientos que se obtuvieron mediante reuniones con el cliente. Cada requerimiento tiene una identificación y un objetivo a cumplir. Además, existen requerimientos que son dependientes entre sí, los cuales cumplen con una función en común.

Requerimientos funcionales

Dentro del sistema se gestionan actividades como: ingreso, actualización, eliminación, consulta y validación de datos, estos son requerimientos que el usuario requiere a diario, los cuales deben estar bien especificados para que el sistema funcione correctamente según las exigencias respecto al software que tiene el cliente.

A continuación, se describen algunos requerimientos funcionales implementados en el sistema:

Requerimiento 1

Tabla 4. Descripción del requerimiento de autenticación de usuarios

Identificación:	RF01
Nombre del Requerimiento:	Autenticación de usuarios.
Características:	Los usuarios para acceder a cualquier funcionalidad del sistema deberán hacerlo mediante un ingreso de usuario.
Descripción:	Dependiendo los privilegios de usuario que tenga podrá acceder a las diferentes funcionalidades del sistema.
Requerimiento NO funcional:	RNF03 (Seguridad)
Prioridad del requerimiento: Alta	

Nota: Detalles del requerimiento de autenticación de usuarios.

La tabla No. 4 contiene información del requerimiento de autenticación de usuarios, el acceso al sistema será mediante un ingreso con usuario y contraseña. Tiene una prioridad

alta porque mediante este requerimiento se precautela la seguridad del sistema. Este requerimiento se enfoca en habilitar los privilegios del usuario mediante el acceso.

Requerimiento 2

Tabla 5. Descripción del requerimiento para ingresar productos

Identificación:	RF02
Nombre del Requerimiento:	Ingresar productos
Características:	Se podrá almacenar toda la información de los productos con sus determinadas características.
Descripción:	Se diferenciará cada producto mediante su código de barras, se los podrá clasificar según su grupo de pertenencia.
Requerimiento NO funcional:	RNF04 (Validación de información)
Prioridad del requerimiento: Alta	

Nota: Detalles del requerimiento para ingresar productos.

La tabla No. 5 contiene información del requerimiento para ingresar productos, este requerimiento será una actividad a realizarse diariamente. Este requerimiento tiene una prioridad alta, porque busca tener integridad en la información. Además, esta información deberá tener un trato especial.

Requerimiento 3

Tabla 6. Descripción del requerimiento para consultar productos

Identificación:	RF03
Nombre del Requerimiento:	Consultar productos
Características:	Se podrá consultar toda la información de los productos con sus determinadas características.
Descripción:	Se muestra información sobre los productos: su precio, color, marca, stock, etc.
Requerimiento NO funcional:	RNF02 (Facilidad de uso)
Prioridad del requerimiento: Alta	

Nota: Detalles del requerimiento para consultar productos.

La tabla No. 6 contiene información del requerimiento para consultar productos, este requerimiento deberá mostrar información confiable para la persona que lo requiera. Tiene una prioridad alta, porque los datos que se consulten serán transmitidos al cliente. Siempre deberá estar disponible esta información.

Requerimiento 4

Tabla 7. Descripción del requerimiento para actualizar productos

Identificación:	RF04
Nombre del Requerimiento:	Actualizar productos
Características:	Se podrá actualizar la información de los productos con sus determinadas características.
Descripción:	Se mantendrá actualizado el stock, si se hace un nuevo pedido se lo registra inmediatamente.
Requerimiento NO funcional:	RNF04 (Validación de información)
Prioridad del requerimiento: Alta	

Nota: Detalles del requerimiento para actualizar productos.

La tabla No. 7 contiene información del requerimiento para actualizar productos, este requerimiento busca manipular la información de los productos, de manera que se actualicen sus datos si es que es necesario. La prioridad es alta, porque los datos que se actualicen se registraron en el inventario.

Requerimiento 5

Tabla 8. Descripción del requerimiento para eliminar productos

Identificación:	RF05
Nombre del Requerimiento:	Eliminar productos
Características:	Se podrá eliminar la información de los productos con sus determinadas características.
Descripción:	Se mantendrá actualizado el catálogo de productos.
Requerimiento NO funcional:	RNF04 (Validación de información)
Prioridad del requerimiento: Alta	

Nota: Detalles del requerimiento para eliminar productos.

La tabla No. 8 contiene información del requerimiento para eliminar productos, este requerimiento permite eliminar aquellos productos que deben quedar fuera de stock por algún motivo. Su prioridad es alta, porque la información que se elimine será sacada del inventario.

Requerimiento 6

Tabla 9. Descripción del requerimiento para ingresar clientes

Identificación:	RF06
Nombre del Requerimiento:	Ingresar clientes
Características:	Se podrá almacenar toda la información de los clientes con sus determinadas características.
Descripción:	Se tendrá información oportuna y confiable que permitirá contactarme con los clientes.
Requerimiento NO funcional:	RNF04 (Validación de información)
Prioridad del requerimiento: Alta	

Nota: Detalles del requerimiento para ingresar clientes.

La tabla No. 9 contiene información del requerimiento para ingresar clientes, este requerimiento será una actividad a realizarse con mucha frecuencia. Este requerimiento tiene una prioridad alta, porque busca tener un listado de clientes confiable. El registro de clientes debe tener datos de contacto.

Requerimiento 7

Tabla 10. Descripción del requerimiento para consultar clientes

Identificación:	RF07
Nombre del Requerimiento:	Consultar clientes
Características:	Se podrá consultar toda la información de los clientes con sus determinadas características.
Descripción:	Se muestra información sobre los clientes: su nombre, teléfono, dirección, etc.
Requerimiento NO funcional:	RNF02 (Facilidad de uso)

Prioridad del requerimiento: Alta
--

Nota: Detalles del requerimiento para consultar clientes.

La tabla No. 10 contiene información del requerimiento para consultar clientes, este requerimiento deberá mostrar información confiable para la persona que lo requiera. Tiene una prioridad alta, porque los datos que se consulten permitirán establecer contacto con los clientes.

Requerimiento 8

Tabla 11. Descripción del requerimiento para actualizar clientes

Identificación:	RF08
Nombre del Requerimiento:	Actualizar clientes
Características:	Se podrá actualizar la información de los clientes con sus determinadas características.
Descripción:	Se mantendrá actualizado el listado de clientes.
Requerimiento NO funcional:	RNF04 (Validación de información)
Prioridad del requerimiento:	Alta

Nota Detalles del requerimiento para actualizar clientes.

La tabla No. 11 contiene información del requerimiento para actualizar clientes, este requerimiento busca manipular la información de los clientes, de manera que se actualicen sus datos si es que es necesario. La prioridad es alta, porque estos datos son muy importantes para contactar clientes.

Requerimiento 9

Tabla 12. Descripción del requerimiento para eliminar clientes

Identificación:	RF09
Nombre del Requerimiento:	Eliminar clientes
Características:	Se podrá eliminar la información de los clientes con sus determinadas características.

Descripción:	Se mantendrá actualizado el listado de clientes, si hay un nuevo cliente se lo registra inmediatamente.
Requerimiento NO funcional:	RNF04 (Validación de información)
Prioridad del requerimiento: Alta	

Nota: Detalles del requerimiento para eliminar clientes.

La tabla No. 12 contiene información del requerimiento para eliminar clientes, este requerimiento permite eliminar aquellos clientes que ya no tienen relación con la distribuidora. Su prioridad es alta, porque la información que se elimine ya no podrá ser usada.

Requerimiento 10

Tabla 13. Descripción del requerimiento para ingresar proveedores

Identificación:	RF10
Nombre del Requerimiento:	Ingresar proveedores
Características:	Se podrá almacenar toda la información de los proveedores con sus determinadas características.
Descripción:	Se tendrá información oportuna y confiable que permitirá contactarme con los proveedores.
Requerimiento NO funcional:	RNF04 (Validación de información)
Prioridad del requerimiento: Alta	

Nota: Detalles del requerimiento para ingresar proveedores.

La tabla No. 13 contiene información del requerimiento para ingresar proveedores, este requerimiento será una actividad a realizarse constantemente. Este requerimiento tiene una prioridad alta, porque busca tener un listado de proveedores confiable. Esto permitirá solicitar oportunamente pedidos.

Requerimiento 11

Tabla 14. Descripción del requerimiento para consultar proveedores

Identificación:	RF11
------------------------	------

Nombre del Requerimiento:	Consultar proveedores
Características:	Se podrá consultar toda la información de los proveedores con sus determinadas características.
Descripción:	Se muestra información sobre los proveedores: su nombre, teléfono, dirección, etc.
Requerimiento NO funcional:	RNF02 (Facilidad de uso)
Prioridad del requerimiento: Alta	

Nota: Detalles del requerimiento de consultar proveedores.

La tabla No. 14 contiene información del requerimiento de consultar proveedores, este requerimiento deberá mostrar información confiable para la persona que lo requiera. Tiene una prioridad alta, porque los datos que se consulten permitirán realizar pedidos a los proveedores.

Requerimiento 12

Tabla 15. Descripción del requerimiento para actualizar proveedores

Identificación:	RF12
Nombre del Requerimiento:	Actualizar proveedores
Características:	Se podrá actualizar la información de los proveedores con sus determinadas características.
Descripción:	Se mantendrá actualizado el listado de proveedores, si hay un nuevo proveedor se lo registra inmediatamente.
Requerimiento NO funcional:	RNF04 (Validación de información)
Prioridad del requerimiento: Alta	

Nota: Detalles del requerimiento para actualizar proveedores.

La tabla No. 15 contiene información del requerimiento para actualizar proveedores, este requerimiento busca manipular la información de los proveedores, de manera que se actualicen sus datos si es que es necesario. La prioridad es alta, porque estos datos son muy importantes para contactar a los proveedores.

Requerimiento 13

Tabla 16. Descripción del requerimiento para eliminar proveedores

Identificación:	RF13
Nombre del Requerimiento:	Eliminar proveedores
Características:	Se podrá eliminar la información de los proveedores con sus determinadas características.
Descripción:	Se mantendrá actualizado el listado de proveedores.
Requerimiento NO funcional:	RNF04 (Validación de información)
Prioridad del requerimiento: Alta	

Nota: Detalles del requerimiento para eliminar proveedores.

La tabla No. 16 contiene información del requerimiento para eliminar proveedores, este requerimiento permite eliminar aquellos clientes que ya no tienen relación con la distribuidora. Su prioridad es alta, porque la información que se elimine ya no podrá ser usada.

Requerimientos no funcionales

Son requerimientos generales, se enfocan en las propiedades que debe tener el sistema. Estos requerimientos son aplicables para todo el sistema. Dentro de la clasificación de los requerimientos siempre se debe identificar el propósito del mismo. También se puede dar el caso de que un requerimiento funcional dependa de un requerimiento no funcional.

A continuación, se describen algunos requerimientos no funcionales implementados en el sistema:

Requerimiento 1

Tabla 17. Descripción del requerimiento de escalabilidad del sistema

Identificación:	RNF01
Nombre del Requerimiento:	Escalabilidad
Características:	El sistema deberá ser flexible para agregar nuevos módulos.
Descripción:	Si es necesario en un futuro se podrá expandir las funcionalidades del sistema.
Prioridad del requerimiento: Alta	

Nota: Detalles del requerimiento de escalabilidad.

La tabla No. 17 contiene información del requerimiento de escalabilidad, este requerimiento busca que el sistema se pueda acoplar con otros sistemas o módulos. De igual manera debe estar preparado para un incremento de usuarios y poder manejar una gran diversidad de productos.

Requerimiento 2

Tabla 18. Descripción del requerimiento de facilidad de uso del sistema

Identificación:	RNF02
Nombre del Requerimiento:	Facilidad de uso
Características:	El sistema tendrá una ayuda para los usuarios.
Descripción:	Se realizará una capacitación y se podrá acceder a un manual de usuario.
Prioridad del requerimiento: Alta	

Nota: Detalles del requerimiento de facilidad de uso.

La tabla No. 18 contiene información del requerimiento de facilidad de uso, este requerimiento busca que el usuario no tenga complicaciones para manipular el sistema. Esto evitará problemas y reducirá el tiempo de capacitación a los usuarios. Reducirá los índices de cometer errores por parte de los usuarios.

Requerimiento 3

Tabla 19. Descripción del requerimiento de seguridad del sistema

Identificación:	RNF03
Nombre del Requerimiento:	Seguridad
Características:	Ventana de acceso
Descripción:	Mediante un usuario y contraseña se podrá acceder al sistema.
Prioridad del requerimiento: Alta	

Nota: Detalles del requerimiento de seguridad.

La tabla No. 19 contiene información del requerimiento de seguridad, este requerimiento busca que el sistema esté protegido. Porque solamente podrán acceder a él los usuarios

registrados y con permisos. Su prioridad es alta, porque de este requerimiento depende el cuidado de la información.

Requerimiento 4

Tabla 20. Descripción del requerimiento de validación del sistema

Identificación:	RNF04
Nombre del Requerimiento:	Validación de información
Características:	Control de los datos que van a ser ingresados al sistema.
Descripción:	No se aceptarán campos vacíos (excepciones), se pedirán datos reales, diferenciación entre números y letras.
Prioridad del requerimiento: Alta	

Nota: Detalles del requerimiento de validación.

La tabla No. 20 contiene información del requerimiento de validación, este requerimiento controlara los formularios, específicamente los campos que deben ser llenados o seleccionados por el usuario. Servirá de guía y protección para los datos. Al ser información que será manipulada por varias personas deberá ser información segura.

Requerimiento 5

Tabla 21. Descripción del requerimiento de disponibilidad del sistema

Identificación:	RNF05
Nombre del Requerimiento:	Disponibilidad
Características:	El sistema estará activo siempre que el usuario quiera acceder a el.
Descripción:	Obtención de reportes, Consulta de stock, Actualización de información.
Prioridad del requerimiento: Alta	

Nota: Detalles del requerimiento de disponibilidad.

La tabla No. 21 contiene información del requerimiento de disponibilidad, este requerimiento busca la estabilidad del sistema. Cualquier mantenimiento que se deba hacer, se hará en un horario que no afecte a los usuarios. Su prioridad es alta, porque el sistema maneja un proceso indispensable para el negocio.

Requerimiento 6

Tabla 22. Descripción del requerimiento de eficiencia del sistema

Identificación:	RNF06
Nombre del Requerimiento:	Eficiencia
Características:	El sistema debe cumplir con sus tareas usando el mínimo de recursos
Descripción:	Poco consumo de memoria interna, rapidez en procesamiento de datos.
Prioridad del requerimiento: Alta	

Nota: Detalles del requerimiento de eficiencia.

La tabla No. 22 contiene información del requerimiento de eficiencia, este requerimiento busca que el sistema sea rápido y ligero. Su prioridad es alta, porque tiene que mantener su nivel de respuesta, no perder su calidad, aunque use pocos recursos. Su enfoque es el mínimo consumo de recursos.

Requerimiento 7

Tabla 23. Descripción del requerimiento de eficacia del sistema

Identificación:	RNF07
Nombre del Requerimiento:	Eficacia
Características:	El sistema debe cumplir con sus tareas usando todos los recursos que necesite cuando sea estrictamente necesario.
Descripción:	Lo importante es obtener resultados sin considerar el uso de recursos.
Prioridad del requerimiento: Alta	

Nota: Detalles del requerimiento de eficacia.

La tabla No. 23 contiene información del requerimiento de eficacia, este requerimiento busca que en casos especiales donde lo importante es obtener resultados, el sistema este apto para usar todos sus recursos. Su prioridad es alta, porque el sistema debe estar preparado para enfrentar estas situaciones.

Requerimiento 8

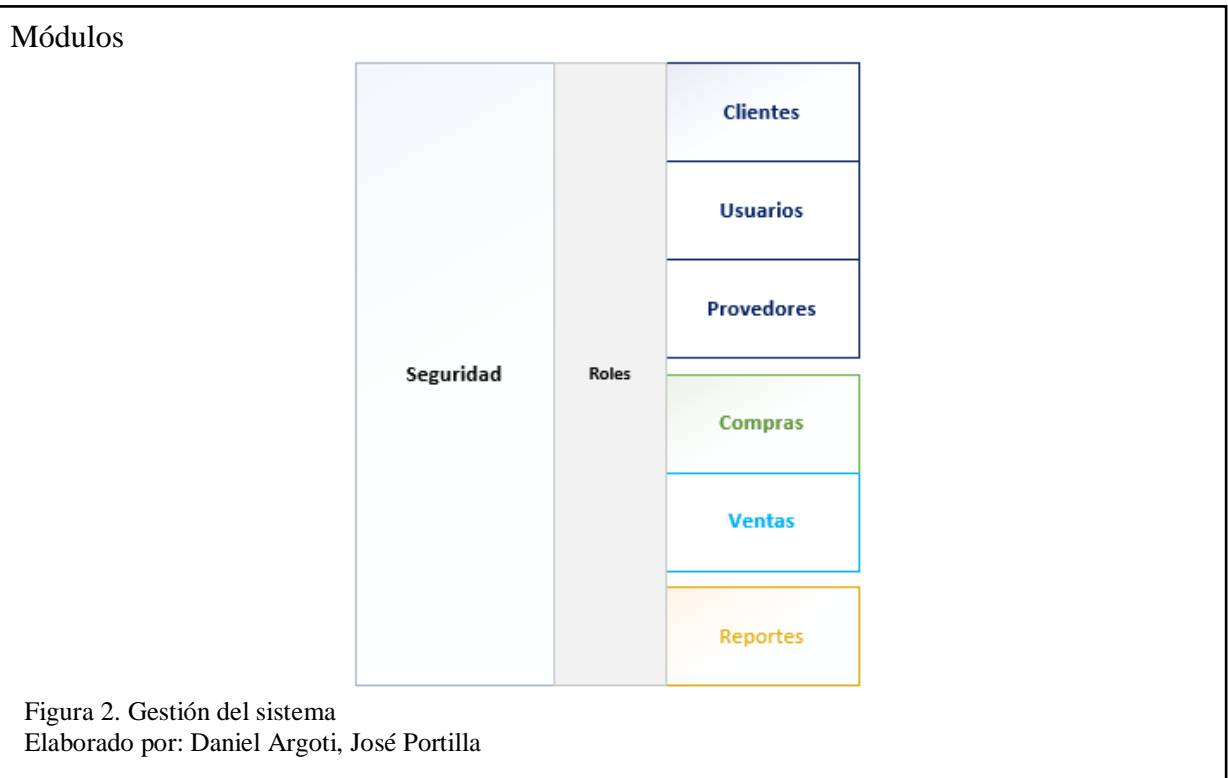
Tabla 24. Descripción del requerimiento de integridad del sistema

Identificación:	RNF08
Nombre del Requerimiento:	Integridad de la información
Características:	El sistema debe precautelar la integridad de los datos.
Descripción:	La información que se muestre y use debe ser confiable y coherente.
Prioridad del requerimiento:	Alta

Nota: Detalles del requerimiento de integridad.

La tabla No. 24 contiene información del requerimiento de integridad, este requerimiento busca que la información que se almacene y deba ser mostrada sea protegida. Su prioridad es alta, porque de la información que se obtenga se tomara decisiones. Se busca garantizar información confiable.

2.1.4 Segmentación del sistema.



En la figura No. 2 se muestra que el sistema se ha dividido en 3 módulos cada uno con su funcionalidad específica, esto permitirá una mejor gestión del sistema porque no habrá conflictos internos. Cada módulo será independiente, pero sí tendrán una interacción entre si cuando lo requieran.

Módulo Clientes

Este módulo gestiona toda la información relacionada a los clientes, esto permite que la distribuidora pueda acceder a esta información en el momento que la requiera. Se maneja datos como número de cédula, nombres, teléfono, dirección. Lo que permitirá que el sistema genere reportes que muestren la información de los clientes que se necesite visualizar. En este módulo se podrá realizar acciones como insertar, eliminar, actualizar y consultar clientes. El acceso a este módulo dependerá de los permisos que se le otorguen a un usuario para que pueda ejecutar estas acciones.

Módulo Proveedores

Este módulo gestiona toda la información relacionada a los proveedores, esto permite que la distribuidora pueda acceder a esta información en el momento que la requiera. Se manejan datos como nombres, número de cédula, teléfono, dirección. El sistema generará reportes que muestren la información de los proveedores que necesitamos visualizar. Esto facilitará el ponerse en contacto con un proveedor para realizar un pedido de productos.

Modulo Productos

Este módulo gestiona toda la información de los productos, se usará un código de barras que permitirá identificar a los productos. Lo que se busca es conocer la situación de cada producto, stock actual, ubicación del producto, precio del producto. Esto disminuirá la

perdida de productos y optimizará el uso de recursos como tiempo y personal. También se registra las entradas y salidas de cada producto para tener un control adecuado de inventarios. Esto permite mantener información actualizada que este siempre disponible cuando se requiera acceder a la misma.

Seguridad

Con el ingreso de usuarios se podrá acceder al sistema, cada usuario tendrá privilegios asignados. Lo que se busca es mantener segura la información que se maneja. Hay que precautelar el acceso al sistema para evitar perdida de datos. Este control se hará con ayuda de un usuario y contraseña. El rol del usuario será asignado por el administrador del sistema según su ocupación dentro de la distribuidora.

2.2 Diseño

2.2.1 Diagramas de casos de uso.

Los diagramas de casos de uso permiten comprender el comportamiento del sistema, se maneja una relación entre algo o alguien y las actividades que hará sobre el sistema. Los casos de uso ayudan a interpretar lo que el usuario quiere que el sistema haga, (Fernandez, 2006).

Caso de uso: acceso al sistema

Caso de uso #1

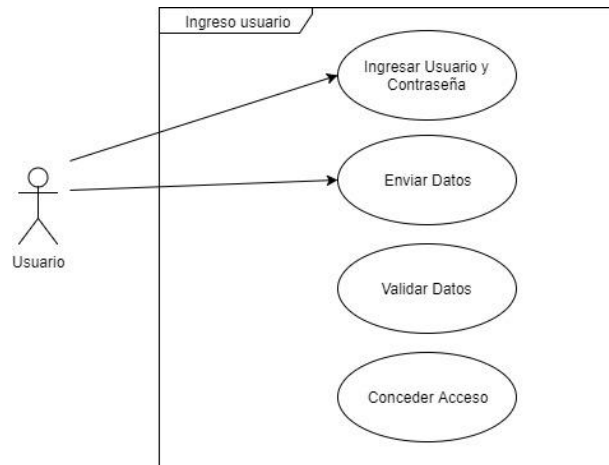


Figura 3. Ingreso de usuario

Elaborado por: Daniel Argoti, José Portilla

En la figura No. 3 se muestra que el usuario para acceder al sistema tiene que ingresar su nombre y contraseña. Estos datos son enviados para que el sistema los verifique y si son correctos se concederá el acceso al sistema. Lo que se busca es que el sistema sea seguro y solo personas autorizadas tengan acceso.

Caso de uso: gestión de productos

Caso de uso #2

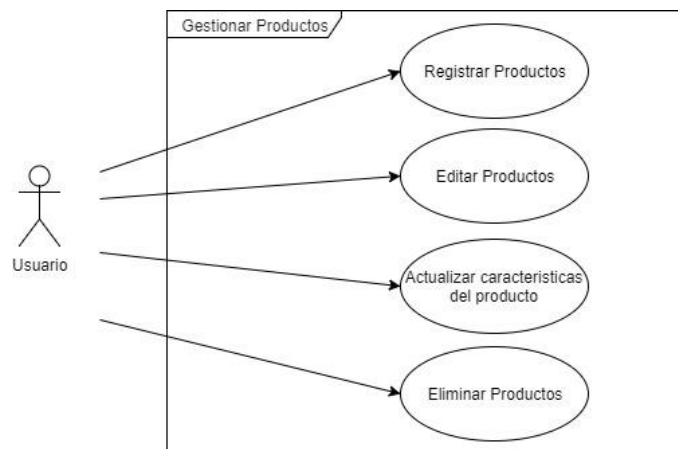
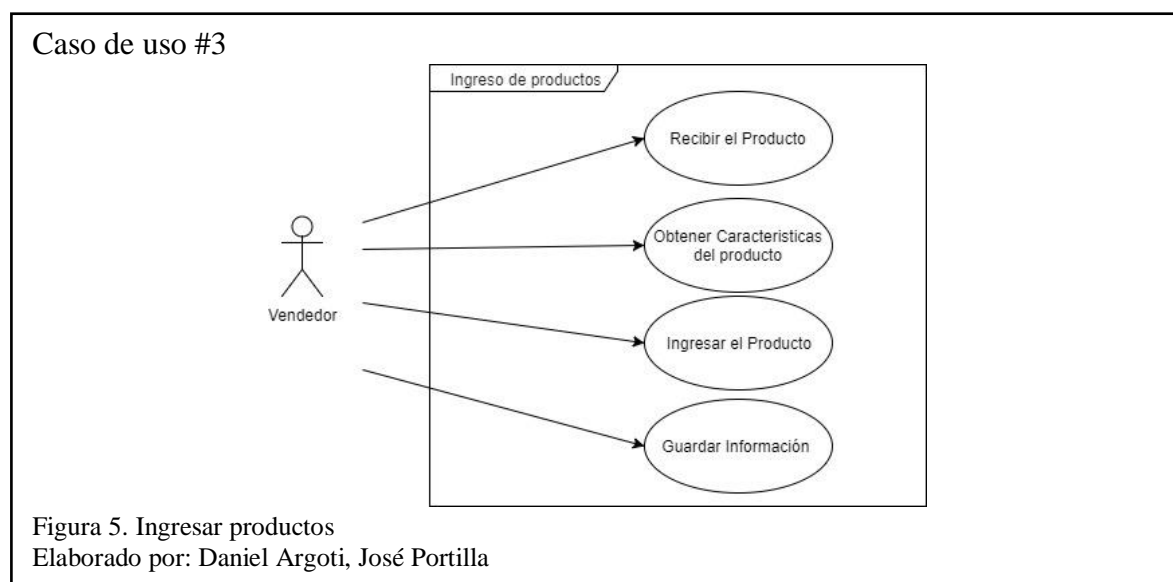


Figura 4. Gestionar productos

Elaborado por: Daniel Argoti, José Portilla

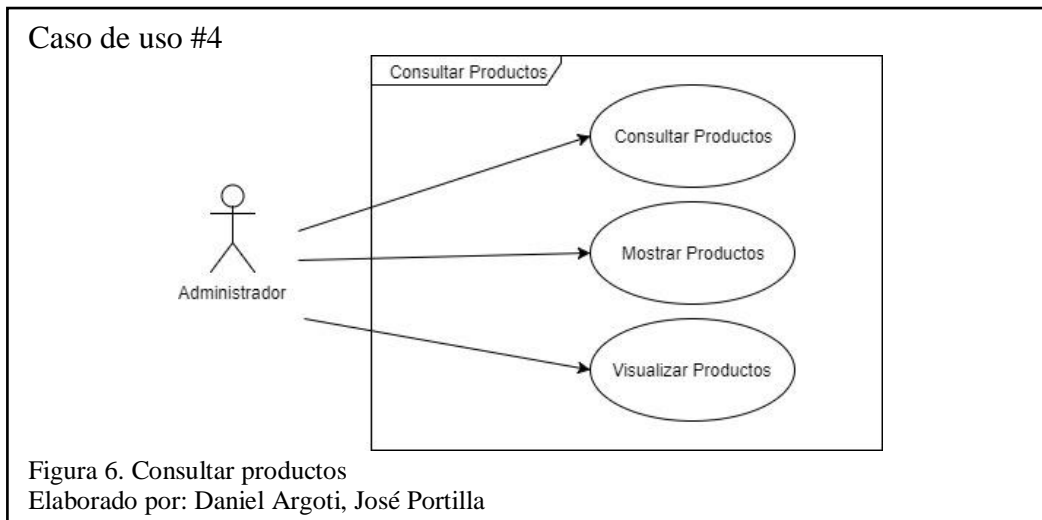
En la figura No. 4 se muestra las opciones generales que tendrá un usuario para gestionar el módulo de productos. Hay 4 opciones a las que puede acceder: ingreso de productos, consulta de productos, actualización de productos y eliminación de productos. El administrador del sistema designará al encargado de cumplir con estas actividades.

Caso de uso: ingreso de productos



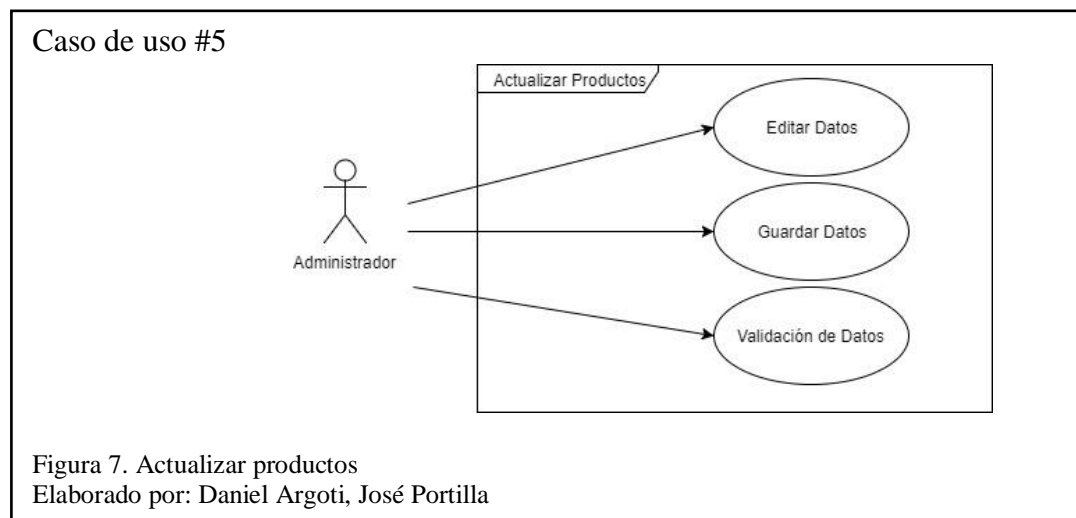
En la figura No. 5 se muestra que el vendedor para poder hacer un ingreso de productos, primero debe recibir la mercadería e identificar sus características. Luego registrar los datos solicitados en un formulario que permitirá guardarlos en el sistema. Esto se hace para las entradas de cada producto.

Caso de uso: consulta de productos



En la figura No. 6 se muestra que el usuario debe acceder a la opción de consulta de productos para poder realizar su búsqueda. Habrá algunos filtros de búsqueda que se podrán usar. Esto se hace para que el usuario pueda buscar información relacionada a los productos en el momento que lo requiera

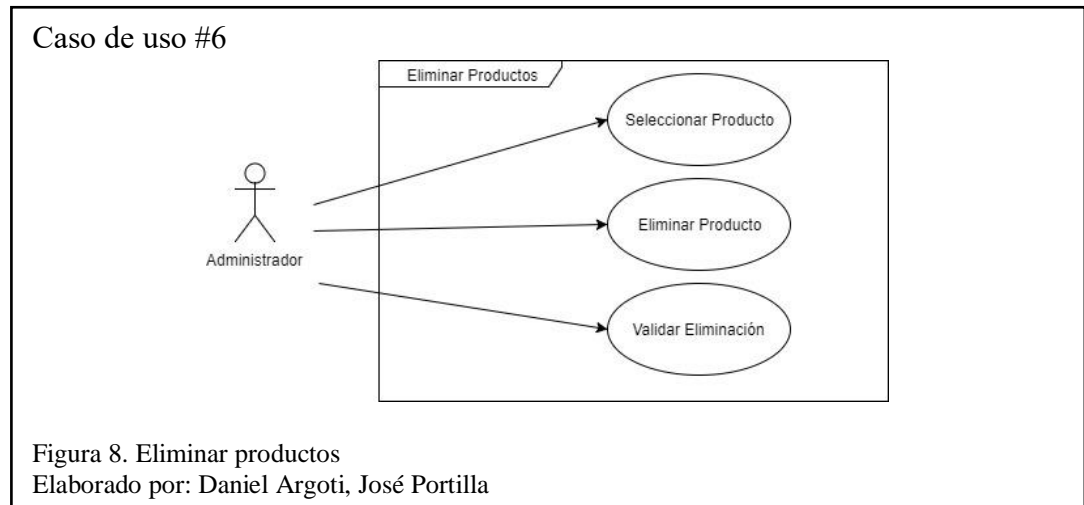
Caso de uso: actualización de productos



En la figura No. 7 se muestra que el usuario puede editar la información ya guardada de un producto. Mediante un formulario de edición se habilitarán los campos que se puedan editar. Porque habrá algunos campos que no sean modificables por su relación interna con

datos ya procesados.

Caso de uso: eliminación de productos



En la figura No. 8 se muestra que el usuario en la opción eliminar debe elegir el producto a eliminar y posteriormente confirmar su eliminación para guardar los cambios. Los cambios efectuados y guardados no se pueden deshacer, por lo que hay que seleccionar correctamente el producto a eliminar.

Caso de uso: salida de productos

Caso de uso #7

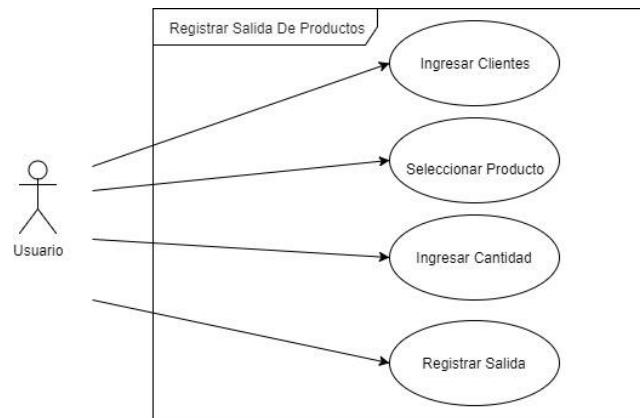


Figura 9. Registro De Salida
Elaborado por: Daniel Argoti, José Portilla

En la figura No. 9 se muestra que el usuario para registrar una salida de productos debe proporcionar al sistema de datos como: el nombre del cliente, el código de barras del producto y la cantidad de salida. Esto permitirá generar un reporte de salida, estos reportes se almacenarán para tener un historial de respaldo.

Caso de uso: gestión de clientes

Caso de uso #8

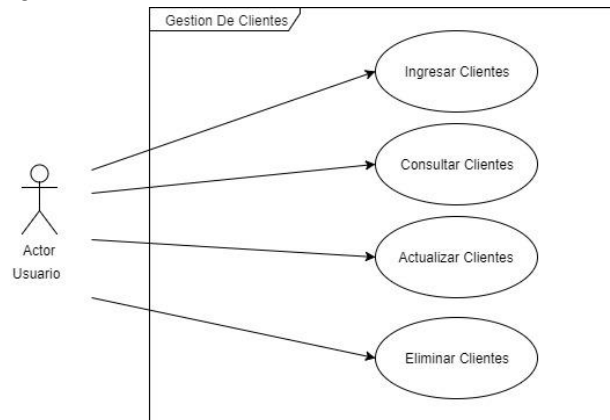
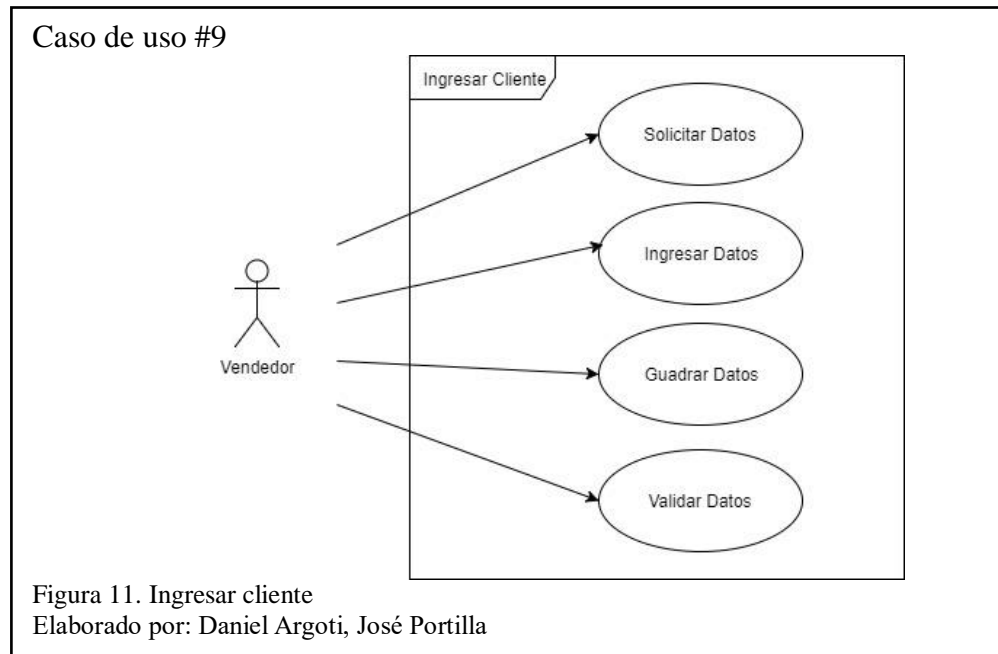


Figura 10. Gestionar clientes
Elaborado por: Daniel Argoti, José Portilla

En la figura No. 10 se muestra las opciones generales que tendrá un usuario para gestionar el módulo de clientes. Hay 4 opciones a las que puede acceder: ingreso de clientes,

consulta de clientes, actualización de clientes y eliminación de clientes. El administrador del sistema designará al encargado de cumplir con estas actividades.

Caso de uso: ingreso de clientes



En la figura No. 11 se muestra que el vendedor para poder hacer un ingreso de clientes, primero debe solicitar los datos del cliente. Luego registrar los datos requeridos en un formulario que permitirá guardarlos en el sistema. Esto se hace para el registro de un nuevo cliente.

Caso de uso: consulta de clientes

Caso de uso #10

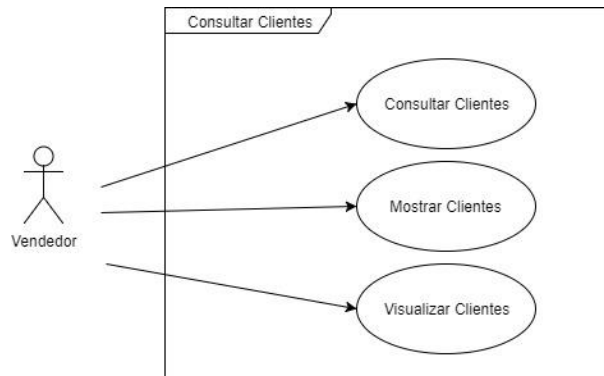


Figura 12. Consultar clientes
Elaborado por: Daniel Argoti, José Portilla

En la figura No. 12 se muestra que el usuario debe acceder a la opción de consulta de clientes para poder realizar su búsqueda. Habrá algunos filtros de búsqueda que se podrán usar. Esto se hace para que el usuario pueda buscar información relacionada a los clientes en el momento que lo requiera

Caso de uso: actualización de clientes

Caso de uso #11

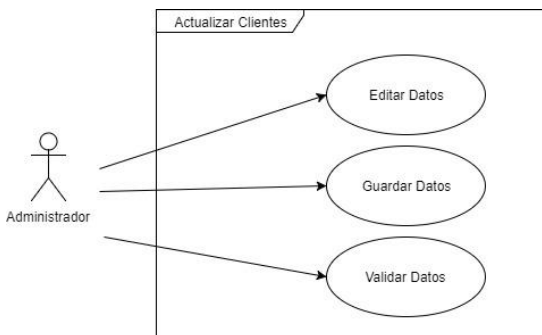
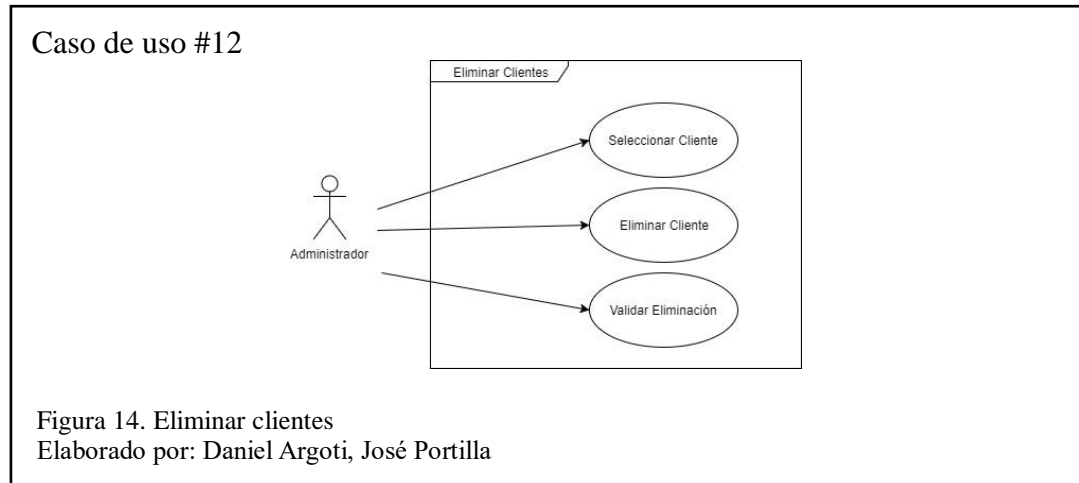


Figura 13. Actualizar clientes
Elaborado por: Daniel Argoti, José Portilla

En la figura No. 13 se muestra que el usuario puede editar la información ya guardada de un cliente. Mediante un formulario de edición se habilitarán los campos que se puedan

editar. Porque habrá algunos campos que no sean modificables por su relación interna con datos ya procesados.

Caso de uso: eliminación de clientes



En la figura No. 14 se muestra que el usuario en la opción eliminar debe elegir el cliente a eliminar y posteriormente confirmar su eliminación para guardar los cambios. Los cambios efectuados y guardados no se pueden deshacer, por lo que hay que seleccionar correctamente el cliente a eliminar.

Caso de uso: gestión de proveedores

Caso de uso #13

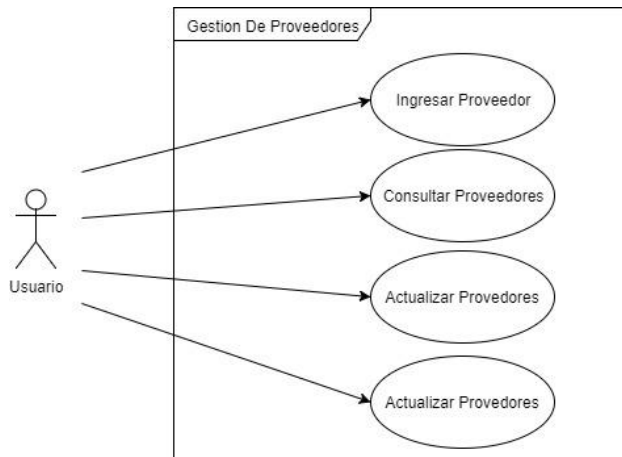


Figura 15. Gestionar proveedores
Elaborado por: Daniel Argoti, José Portilla

En la figura No. 15 se muestra las opciones generales que tendrá un usuario para gestionar el módulo de proveedores. Hay 4 opciones a las que puede acceder: ingreso de proveedores, consulta de proveedores, actualización de proveedores y eliminación de proveedores. El administrador del sistema designará al encargado de cumplir con estas actividades.

Caso de uso: ingreso de proveedores

Caso de uso #14

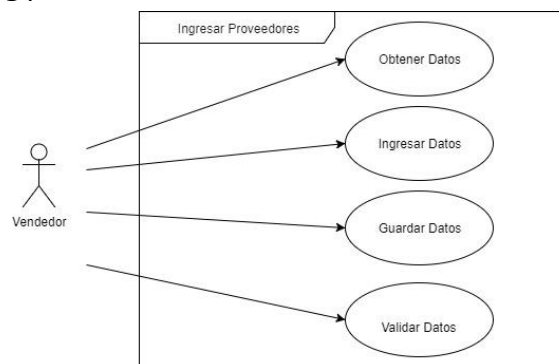
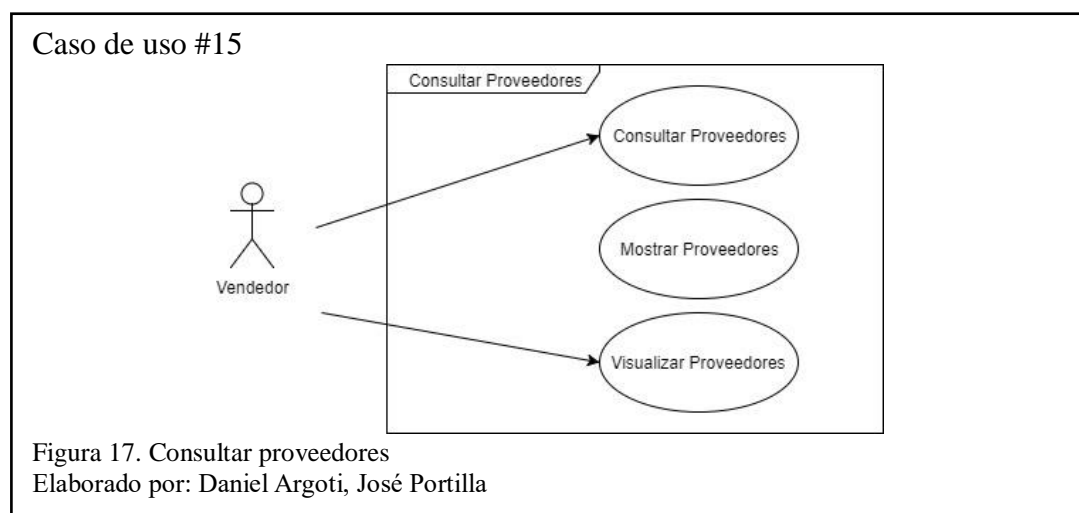


Figura 16. Ingresar proveedores
Elaborado por: Daniel Argoti, José Portilla

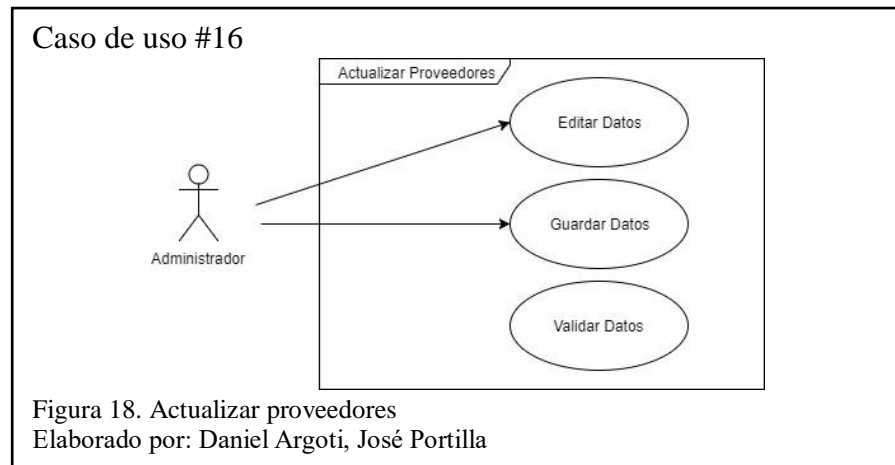
En la figura No. 16 se muestra que el vendedor para poder hacer un ingreso de proveedores primero debe solicitar los datos del proveedor. Luego registrar los datos requeridos en un formulario que permitirá guardarlos en el sistema. Esto se hace para el registro de un nuevo proveedor.

Caso de uso: consulta de proveedores



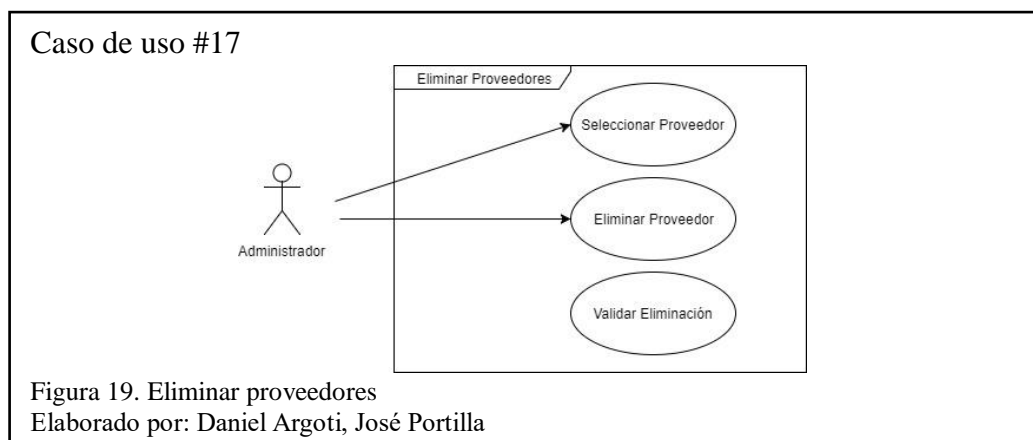
En la figura No. 17 se muestra que el usuario debe acceder a la opción de consulta de proveedores para poder realizar su búsqueda. Habrá algunos filtros de búsqueda que se podrán usar. Esto se hace para que el usuario pueda buscar información relacionada a los proveedores en el momento que lo requiera.

Caso de uso: actualización de proveedores



En la figura No. 18 se muestra que el usuario puede editar la información ya guardada de un proveedor. Mediante un formulario de edición se habilitarán los campos que se puedan editar. Porque habrá algunos campos que no sean modificables por su relación interna con datos ya procesados.

Caso de uso: eliminación de proveedores

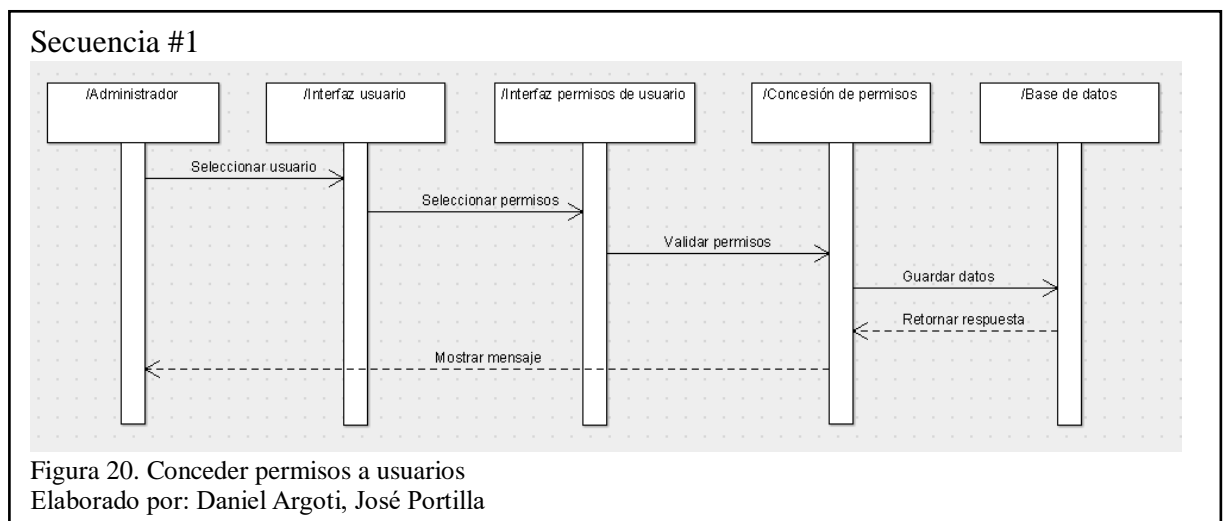


En la figura No. 19 se muestra que el usuario en la opción eliminar debe elegir el proveedor a eliminar y posteriormente confirmar su eliminación para guardar los cambios. Los cambios efectuados y guardados no se pueden deshacer, por lo que hay que seleccionar correctamente el proveedor a eliminar.

2.2.2 Diagramas de secuencia.

Estos diagramas muestran la interacción entre los distintos componentes del sistema, envío y recepción de peticiones. Las acciones detalladas en estos diagramas son secuencias cronológicas que deben cumplirse para obtener el resultado esperado. Brindan una visión clara de cómo debe funcionar el sistema. (Sommerville, 2005)

Diagrama de secuencia para concesión de permisos a usuarios



En la figura No. 20 se muestra la interacción que debe realizar el administrador para asignar permisos a los usuarios, el administrador selecciona al usuario y asigna permisos desde la interfaz de usuarios. Internamente hay una conexión a la base de datos para guardar y mantener los cambios. Se mostrará un mensaje que indique si se ha cumplido o no con la petición.

Diagrama de secuencia para acceso al sistema

Secuencia #2

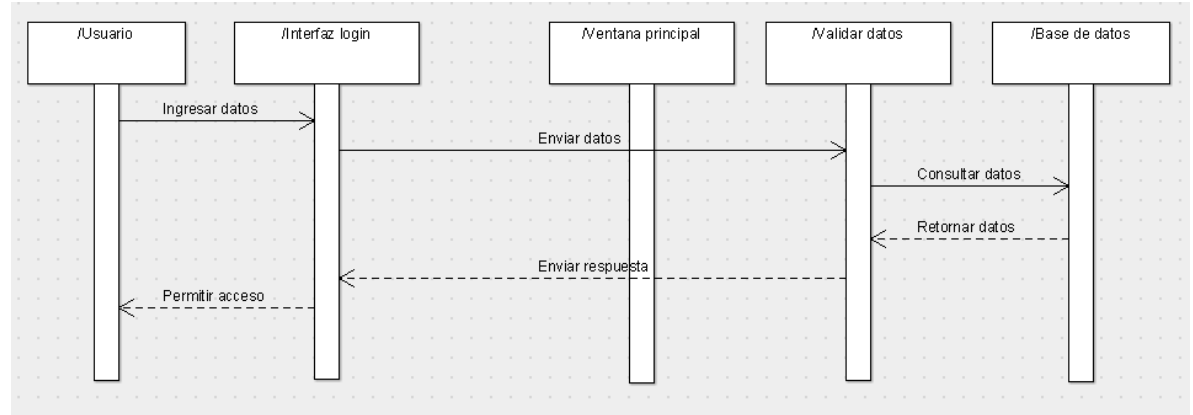


Figura 21. Acceder al sistema

Elaborado por: Daniel Argoti, José Portilla

En la figura No. 21 se muestra la interacción que debe realizar un usuario para acceder al sistema mediante un ingreso, el usuario ingresa su nombre y contraseña desde la interfaz de ingreso. Internamente hay una conexión a la base de datos para verificar y validar los datos. Se mostrará si se tiene acceso al sistema o no, dándole acceso a la ventana principal en donde están los módulos.

Diagrama de secuencia para inserción de datos

Secuencia #3

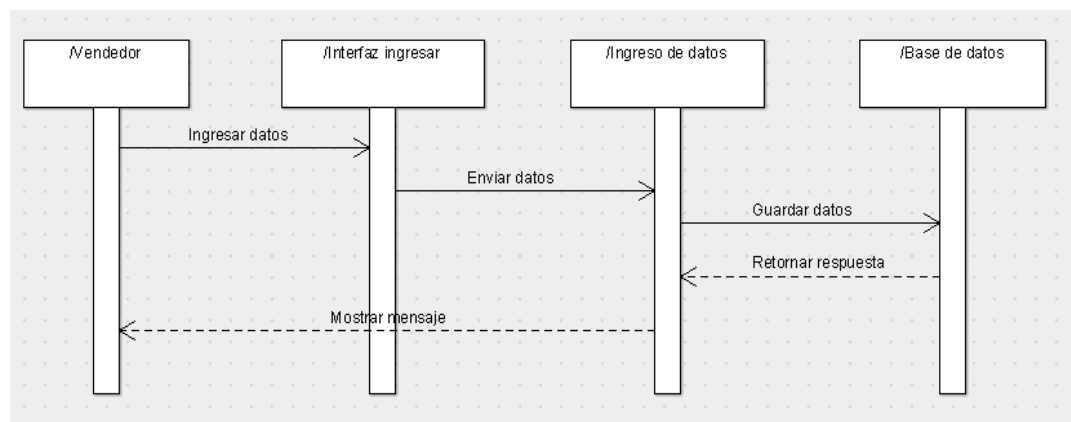
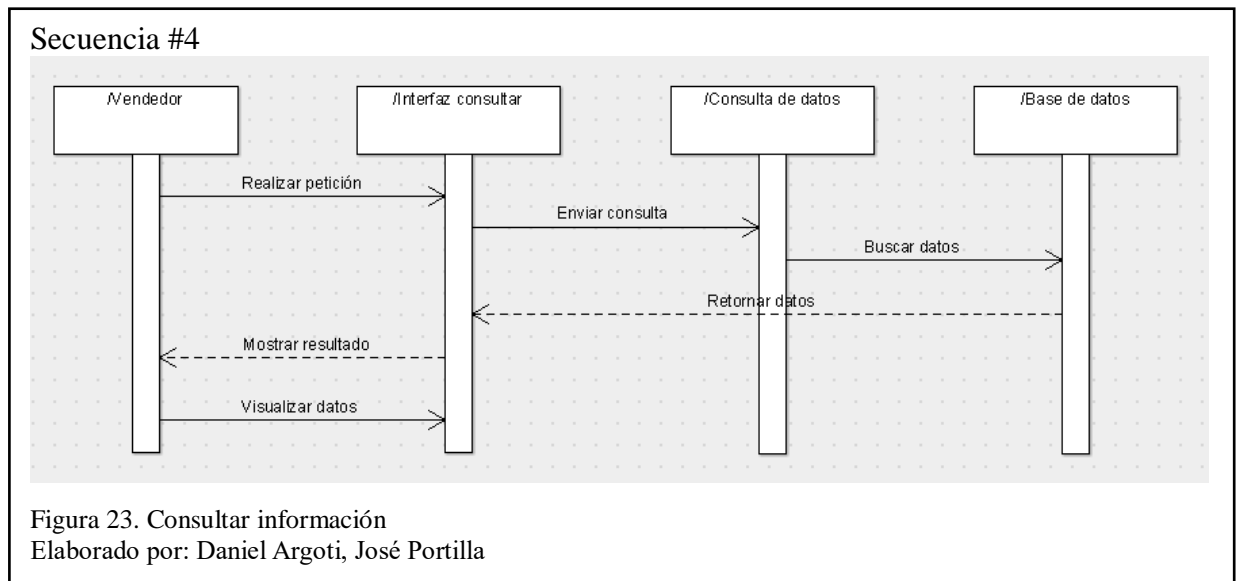


Figura 22. Insertar datos

Elaborado por: Daniel Argoti, José Portilla

En la figura No. 22 se muestra la interacción que debe realizar el vendedor para gestionar el ingreso de datos, el usuario registra los datos desde la interfaz ingresar. Internamente hay una conexión a la base de datos para validar y guardar los datos. Se mostrará un mensaje que indique si se ha cumplido o no con la petición.

Diagrama de secuencia para consulta de información



En la figura No. 23 se muestra la interacción que debe realizar el vendedor para gestionar la consulta de datos, el usuario busca los datos desde la interfaz consultar. Internamente hay una conexión a la base de datos para validar y buscar los datos. Se mostrará un mensaje que indique si se ha cumplido o no con la petición.

Diagrama de secuencia para actualización de información

Secuencia #5

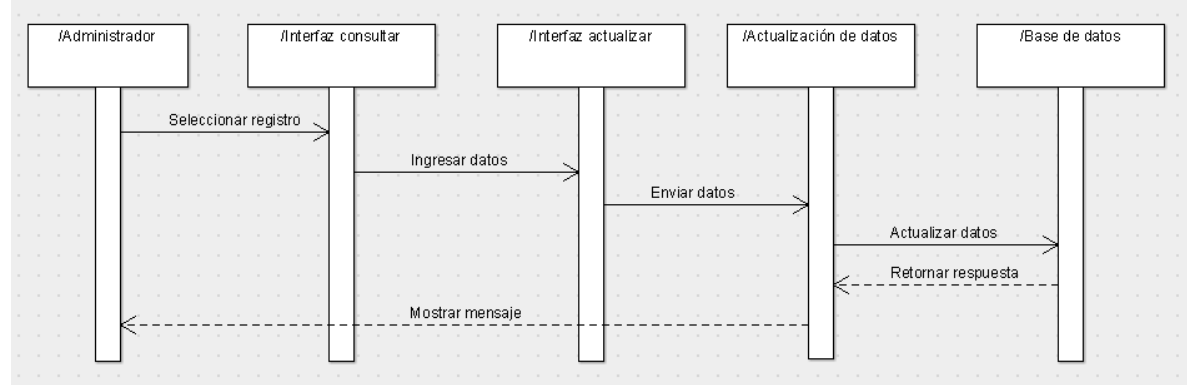


Figura 24. Actualizar información

Elaborado por: Daniel Argoti, José Portilla

En la figura No. 24 se muestra la interacción que debe realizar el administrador para gestionar la actualización de datos, el usuario selecciona el registro y actualiza los datos desde la interfaz actualizar. Internamente hay una conexión a la base de datos para validar y actualizar los datos. Se mostrará un mensaje que indique si se ha cumplido o no con la petición.

Diagrama de secuencia para eliminación de información

Secuencia #6

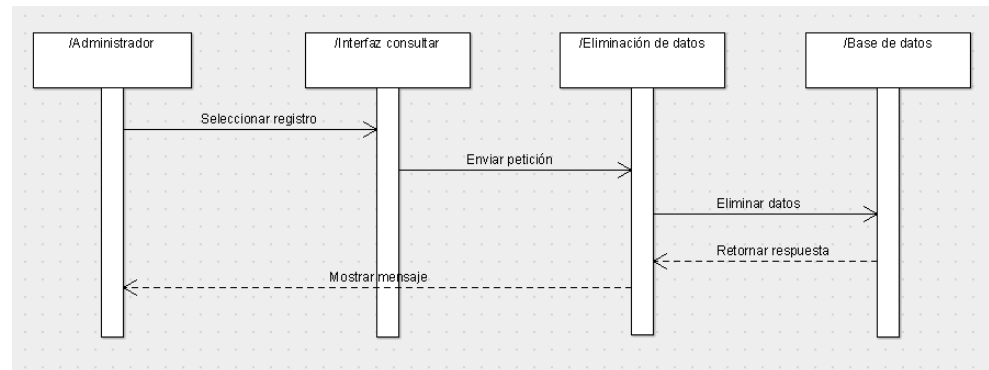


Figura 25. Eliminar información

Elaborado por: Daniel Argoti, José Portilla

En la figura No. 25 se muestra la interacción que debe realizar el administrador para gestionar la eliminación de datos, el usuario selecciona el registro y elimina los datos desde la interfaz eliminar. Internamente hay una conexión a la base de datos para validar y eliminar los datos. Se mostrará un mensaje que indique si se ha cumplido o no con la petición.

Según la metodología OOHDM siguiendo el proceso de diseño se obtendrán los siguientes resultados:

2.2.3 Diseño conceptual.

En esta etapa lo que se hace es diseñar la estructura que será la base del sistema de inventario, se enfoca en el diagrama de la base de datos y el diagrama de clases. Este diseño muestra la estructura arquitectónica del sistema, para realizarlo se parte de la especificación de requerimientos que proporcionaron en la distribuidora. Es una de las partes más importantes del desarrollo porque la información del inventario de la distribuidora será procesada con ayuda de esta estructura.

Diagrama de base de datos

Modelo BDD

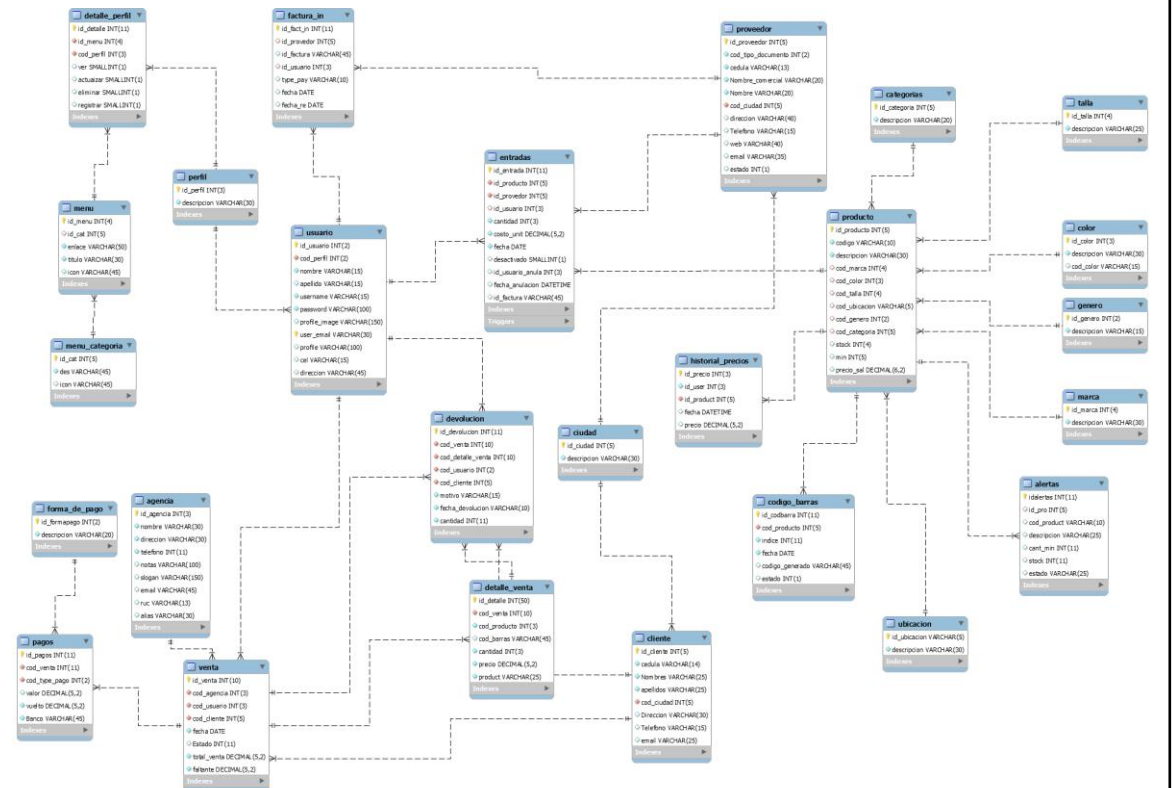


Figura 26. Diagrama BDD
Elaborado por: Daniel Argoti, José Portilla

En la figura No. 26 se muestra el diagrama que contiene un conjunto de tablas que se relacionan entre sí, la tabla productos tienen varios atributos que la conforman, la tabla usuarios se relacionan con la tabla perfiles para dar accesos al sistema. Se controla las entradas y salidas de los productos, la tabla inventario se relaciona a los clientes que son los que compran productos y a los proveedores que son los que abastecen de productos a la distribuidora.

Diagrama de clases

Modelado

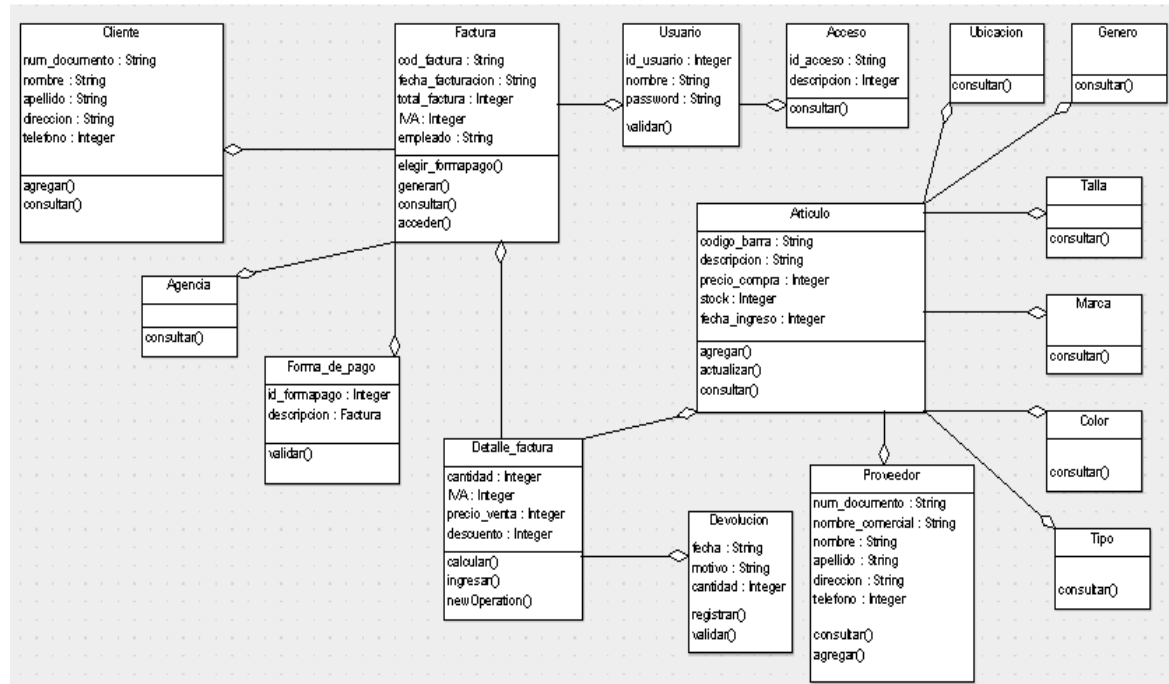
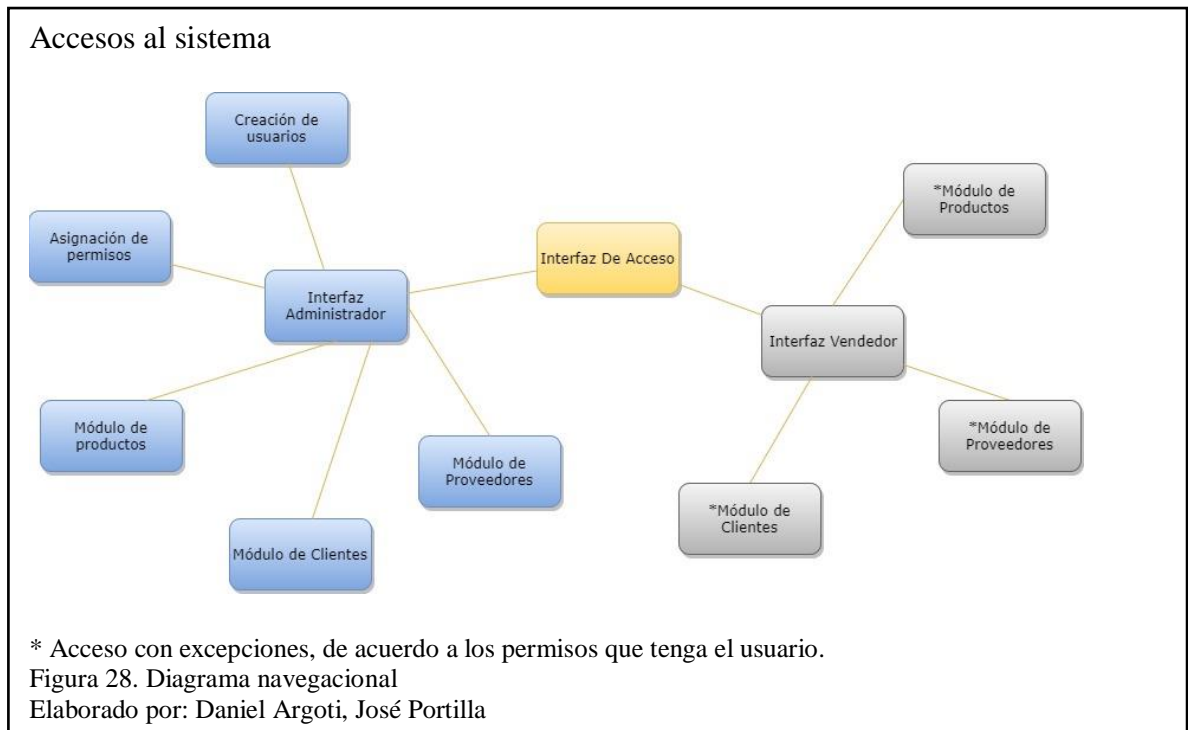


Figura 27. Diagrama de clases
Elaborado por: Daniel Argoti, José Portilla

En la figura No. 27 que se muestra está el diagrama que modela la estructura de las funcionalidades del sistema con cada una de sus características, a partir de aquí se establecerá cuáles serán las ventanas necesarias a ser usadas, cada clase tiene métodos que permitirán procesar la información y manipular el sistema. Además, se diseña el camino a seguir y un orden específico para que el sistema funcione. (Sommerville, 2005)

2.2.4 Diseño navegacional.

En esta etapa lo que se hace es establecer la relación que tendrán las ventanas que representan a cada módulo del sistema. Se crea el diseño que establece como el sistema accederá a cada una de sus interfaces o cada uno de sus módulos. Los menús y submenús tendrán varias opciones a las que solo podrá acceder el usuario que tenga privilegios para hacerlo. El acceso a las interfaces tendrá que seguir este camino de navegación.

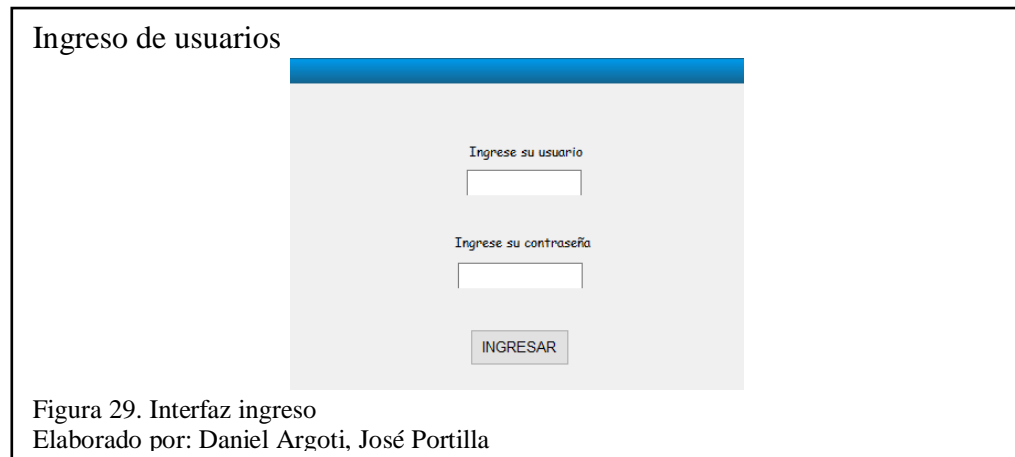


En la figura No. 28 se muestra que la interfaz principal será la interfaz de acceso, hay 2 tipos de acceso como administrador o como vendedor. Según el usuario que accedió al sistema se desplegarán las opciones habilitadas. El encargado de configurar el acceso de navegación es el administrador del sistema.

2.2.5 Diseño de interfaces abstracta.

En esta etapa se diseñan prototipos de las interfaces, esto se hace tratando de emular lo que el cliente quiere. Son prototipos que se enfocan en identificar la idea y buscar un formato común para que los empleados de la distribuidora se acostumbren a usarlo y no tengan problemas al momento de interactuar con el sistema. Además, sirven de ejemplo para mostrarle al cliente y determinar si se ha comprendido su idea, porque hacer cambios en estos prototipos es preferible a hacerlo en una ventana ya construida.

Prototipo ingreso



En la figura No. 29 se muestra que la interfaz de ingreso principalmente tendrá dos campos de texto para completar, uno para ingresar el usuario y otro para ingresar la contraseña. Al hacer clic en el botón de ingreso se capturarán estos datos y si son válidos se podrá acceder a la siguiente ventana, caso contrario se deberá nuevamente ingresar los datos.

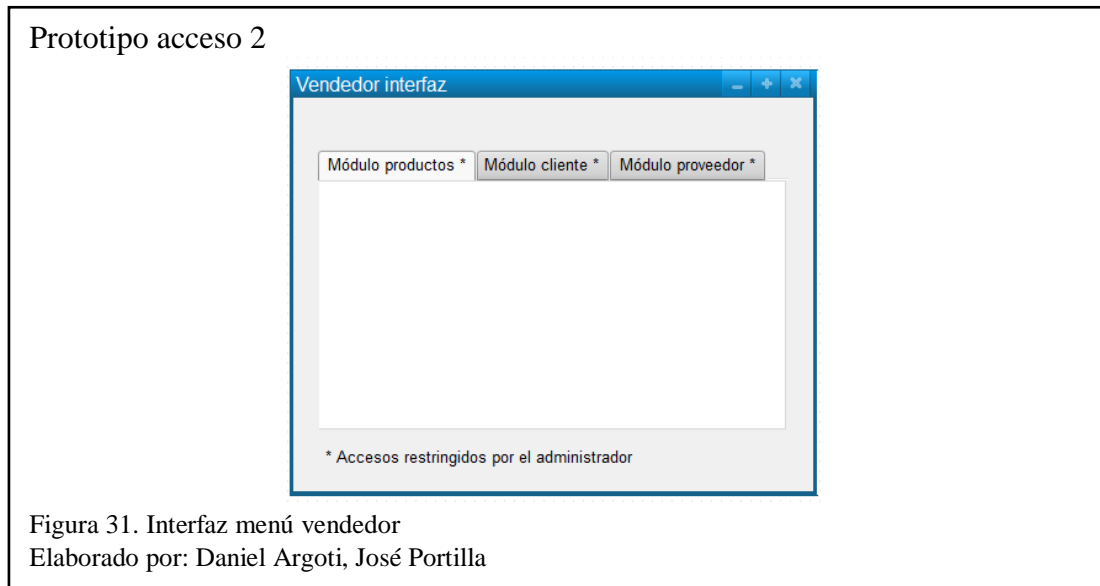
Prototipo interfaz administrador



En la figura No. 30 se muestra que si el usuario se autentico como administrador se desplegará un menú con privilegios totales, donde podrá elegir entre varias opciones para acceder a distintos módulos. Además, tendrá opciones para crear usuarios y asignar

permisos de acceso. Este menú será de fácil acceso y desplegará ventanas nuevas cuando sea necesario.

Prototipo interfaz vendedor



En la figura No. 31 se muestra que si el usuario se autentico como vendedor su nivel de acceso al menú dependerá de sus privilegios. El administrador será el encargado de determinar sus roles. El menú principal de igual manera mostrará los 3 módulos, a diferencia del menú de administrador ya no se verá la gestión de usuarios. Las ventanas también se desplegarán al elegir una opción o abrir un módulo.

Prototipo interfaz módulos

Ventana estándar

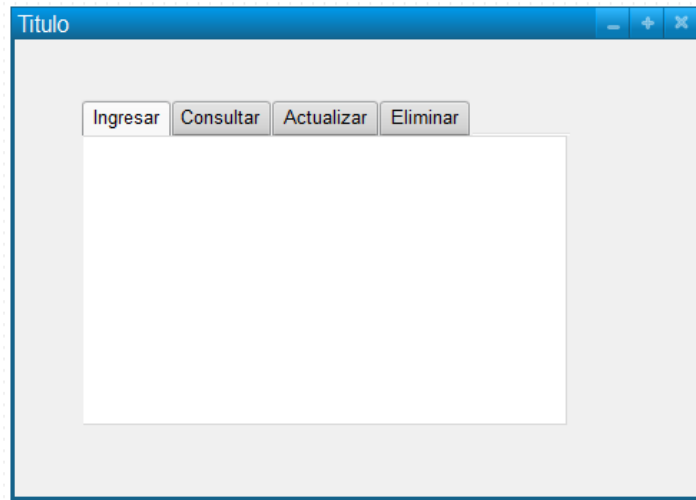


Figura 32. Interfaz productos
Elaborado por: Daniel Argoti, José Portilla

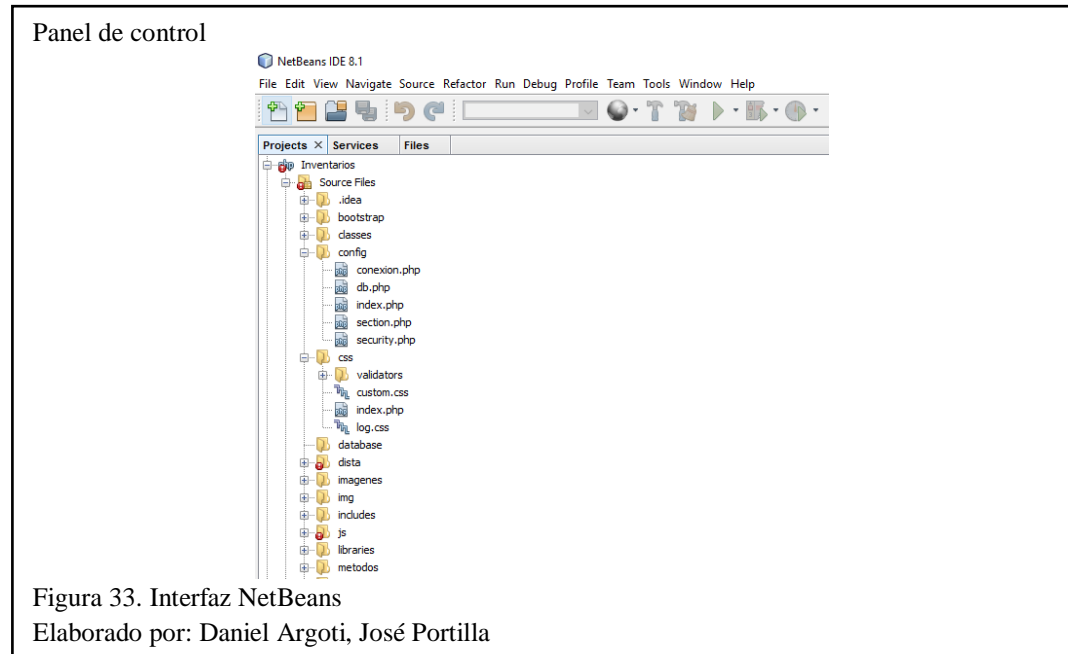
En la figura No. 32 se muestra que dentro de la ventana productos habrá un submenú en el que cada opción me desplegará una ventana distinta: en el ingreso se mostrará un formulario de ingreso, en la actualización se mostrará un formulario con campos específicos, la consulta se desplegará una lista con los datos existentes y en la eliminación se dará a seleccionar un registro a eliminar.

Capítulo 3

Construcción del Sistema

3.1 Configuración de herramientas de desarrollo

NetBeans



En la figura No. 33 se muestra que en el entorno de desarrollo se creó un proyecto de nombre “Sistema de inventarios”, al cual se le agrego clases que representan a cada interfaz, también se añadió librerías que permiten usar recursos necesarios para la creación del sistema. Se agregaron ficheros de configuración para la conexión del sistema con la base de datos y el servidor local.

Xampp

Panel de control

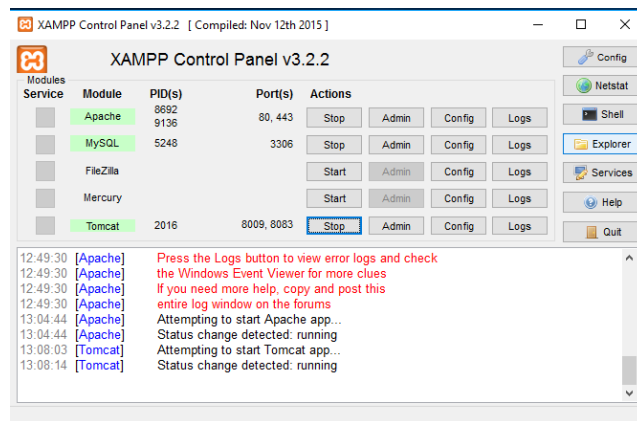


Figura 34. Interfaz xampp
Elaborado por: Daniel Argoti, José Portilla

En la figura No. 34 se muestra cómo se estableció el servidor local para que se conecte con la base de datos MySQL mediante el acceso al sistema que se subió al servidor. Xampp se dio inicio habilitando sus puertos para tener acceso desde cualquier computador que conozca la dirección exacta y tenga habilitados los permisos.

MySQL

Panel de control

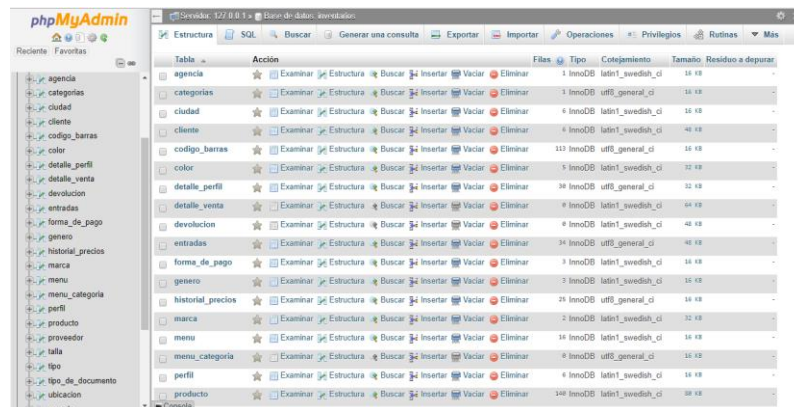


Figura 35. Interfaz mysql
Elaborado por: Daniel Argoti, José Portilla

En la figura No. 35 se muestra cómo se creó la base de datos con el nombre “inventario”, se agregó tablas y se estableció relaciones entre ellas. Se configuró las propiedades de la

base de datos para mantenerla segura con ayuda de claves y respaldos, a las que solo el administrador podrá tener acceso para manipularlas.

3.2 Código del sistema

Conexión base de datos



Figura 36. Código bdd

Elaborado por: Daniel Argoti, José Portilla

En la figura No. 36 se muestra que en esta sección de código se parametriza las variables de conexión a la base de datos, se asigna un nombre de alojamiento, el usuario y contraseña, el nombre de la base de datos. Todas estas variables son verificadas en la conexión entre el sistema y MYSQL, lo cual garantizara el acceso.

Control ingreso

Verificar ingreso

```
if (!$this->db_connection->connect_errno) {
    $user_name = $this->db_connection->real_escape_string($_POST['user_name']);
    $sql = "SELECT user_id, user_name, cod_perfil, firstname, user_email, user_password_hash FROM users
    WHERE user_name = '" . $user_name . "' OR user_email = '" . $user_name . "'";
    $sql = "SELECT * FROM usuario WHERE username='" . $user_name . "' OR user_email='" . $user_name . "'";
    $result_of_login_check = $this->db_connection->query($sql);
    $sql_agency = "SELECT * FROM agencia ";
    $result_agency = $this->db_connection->query($sql_agency);
    $result_row_ag = $result_agency->fetch_object();
    // if this user exists
    if ($result_of_login_check->num_rows == 1) {
        $result_row = $result_of_login_check->fetch_object();
        $_SESSION['image'] = $result_row->profile_image;
        $_SESSION['user_email'] = $result_row->user_email;
        //user password hash
        if (password_verify($_POST['user_password'], $result_row->password)) {
            $_SESSION['user_id'] = $result_row->id_usuario;
            $_SESSION['firstname'] = $result_row->nombre;
            $_SESSION['lastname'] = $result_row->apellido;
            $_SESSION['user_name'] = $result_row->username;
            $_SESSION['user_email'] = $result_row->user_email;
            $_SESSION['profile'] = $result_row->profile;
            $_SESSION['cod_perfil'] = $result_row->cod_perfil;
            $_SESSION['user_login_status'] = 1;
            $_SESSION['agency'] = $result_row_ag->nombre;
            $_SESSION['alias_ag'] = $result_row_ag->alias;
        } else {
            $this->errors[] = "Usuario y/o contraseña no coinciden.";
        }
    } else {
        $this->errors[] = "Usuario y/o contraseña no coinciden.";
    }
}
```

Figura 37. Código login

Elaborado por: Daniel Argoti, José Portilla

En la figura No. 37 se muestra que en esta sección de código se controla el acceso al sistema mediante el ingreso, se toman las variables de nombre y usuario que ingresa el usuario y se las compara con la información almacenada en la base de datos. Si los datos coinciden se permite acceder al sistema, caso contrario se mostrará un mensaje de datos incorrectos.

Sesión activa

Verificar acceso

```
<?php
$active_usuarios = "active";
$title = "SISTMA TE O INVENTARIO";
$image_log = $_SESSION['image'];
$id_user = $_SESSION['user_id'];
$firstname = $_SESSION['firstname'];
$lastname = $_SESSION['lastname'];
$user_name = $_SESSION['user_name'];
$user_email = $_SESSION['user_email'];
$profile = $_SESSION['profile'];
$cod_per = $_SESSION['cod_perfil'];
$alias = $_SESSION['alias_ag'];
$name_business = $_SESSION['agency'];

if (!isset($_SESSION['detalle'])) {
    $_SESSION['detalle']=array();
}

if ($profile == "") {
    $profil = "hold-transition skin-blue sidebar-mini";
} else {
    $profil = $profile;
}

?>
```

Figura 38. Código acceso

Elaborado por: Daniel Argoti, José Portilla

En la figura No. 38 se muestra que en esta sección de código se verifica si la sesión del usuario esta activa en cada página, no se podrá usar el link para acceder a una página específica en el sistema, la única opción para acceder es mediante el ingreso. De igual manera se verifica el perfil del usuario para saber sus privilegios. Se obtienen valores como el nombre de usuario, el perfil, el estado de la sesión para validar el acceso.

Inserción de datos

Insertar en base de datos

```
$sql="INSERT INTO products (codigo_producto, nombre_producto, date_added, precio_producto, stock, id_categoria) "
. "VALUES ('$codigo','$nombre','$date_added','$precio_venta', '$stock','$id_categoria)";
$query_new_insert = mysqli_query($con,$sql);
if ($query_new_insert){
    $messages[] = "Producto ha sido ingresado satisfactoriamente.";
    $id_producto=get_row('products','id_producto', 'codigo_producto', $codigo);
    $user_id=$_SESSION['user_id'];
    $firstname=$_SESSION['firstname'];
    $nota="$firstname agregó $stock producto(s) al inventario";
    echo guardar_historial($id_producto,$user_id,$date_added,$nota,$codigo,$stock);
} else{
    $errors []= "Lo siento algo ha salido mal intenta nuevamente.".mysqli_error($con);
}
} else {
    $errors []= "Error desconocido.";
}
```

Figura 39. Código insertar

Elaborado por: Daniel Argoti, José Portilla

En la figura No. 39 se muestra que en esta sección de código se gestiona el ingreso de datos mediante el método de inserción, se toman las variables de código de producto, nombre, fecha de ingreso, precio, stock y categoría que ingresa el vendedor para enviarlas a la base de datos. Si los datos son validados se los almacenará, caso contrario se mostrará un mensaje de datos incorrectos.

Actualización de datos

Actualizar en base de datos

```
$sql="UPDATE products SET codigo_producto='".$codigo."', nombre_producto='".$nombre."', id_categoria='".$categoria."', "
. "precio_producto='".$precio_venta."', stock='".$stock.'" WHERE id_producto='".$id_producto.'";
$query_update = mysqli_query($con,$sql);
if ($query_update){
    $messages[] = "Producto ha sido actualizado satisfactoriamente.";
} else{
    $errors []= "Lo siento algo ha salido mal intenta nuevamente.".mysqli_error($con);
}
} else {
    $errors []= "Error desconocido.";
}
```

Figura 40. Código actualizar

Elaborado por: Daniel Argoti, José Portilla

En la figura No. 40 se muestra que en esta sección de código se gestiona la actualización de datos mediante el método de actualización, se toman las variables de código de producto, nombre, fecha de ingreso, precio, stock y categoría que edita el vendedor para

enviarlas a la base datos. Si los datos son validados se los actualizará, caso contrario se mostrará un mensaje de datos incorrectos.

Eliminación de datos

Eliminar en base de datos

```
if (isset($_GET['id'])) {
    $id_producto = intval($_GET['id']);
    if ($delete = mysqli_query($con, "DELETE FROM products WHERE id_producto='" . $id_producto . "'")) {
        ?>
        <div class="alert alert-success alert-dismissible" role="alert">
            <button type="button" class="close" data-dismiss="alert" aria-label="Close"><span aria-hidden="true">
            <strong>Aviso!</strong> Datos eliminados exitosamente.
        </div>
        <?php
    } else {
        ?>
        <div class="alert alert-danger alert-dismissible" role="alert">
            <button type="button" class="close" data-dismiss="alert" aria-label="Close"><span aria-hidden="true">
            <strong>Error!</strong> Lo siento algo ha salido mal intenta nuevamente.
        </div>
        <?php
    }
}
```

Figura 41. Código eliminar

Elaborado por: Daniel Argoti, José Portilla

En la figura No. 41 se muestra que en esta sección de código se gestiona la eliminación de datos mediante el método de eliminar, se toman las variables de código de producto, que selecciona el vendedor para enviarlas a la base datos. Si los datos son validados se los eliminará, caso contrario se mostrará un mensaje de datos incorrectos.

Consulta de datos

Consultar listado

```
function load(page){
    var q= $("#q").val();
    var id_categoria= $("#id_categoria").val();
    var parametros={ 'action': 'ajax', 'page': page, 'q': q, 'id_categoria': id_categoria };
    $("#loader").fadeIn('slow');
    $.ajax({
        data: parametros,
        url: './ajax/buscar_productos.php',
        beforeSend: function(objeto){
            $('#loader').html(' Cargando...');
        },
        success: function(data){
            $(".outer_div").html(data).fadeIn('slow');
            $('#loader').html('');
        }
    });
}
```

Figura 42. Código consultar

Elaborado por: Daniel Argoti, José Portilla

En la figura No. 42 se muestra que en esta sección de código se gestiona la consulta de información mediante el método de consulta, se toman la variable de código de producto, que selecciona el vendedor para enviarlas a la base datos y buscar el registro que se relacione con ese código. Si los datos son validados se mostrará la información requerida, caso contrario se mostrará un mensaje de datos incorrectos.

Salida de productos

Generar salida

```
for ($i = 1; $i <= count($products); $i++) {
    $cod_product = $products[$i][1];
    $cod_barras = $products[$i][3];
    $cant = $products[$i][2];
    $price = $products[$i][5];
    $product = $products[$i][4];
    $sql_in_sale = "insert into detalle_venta(cod_venta,cod_producto,cod_barras,cantidad,precio,product) values($id
    mysqli_query($con, $sql_in_sale);

    $sql_check_stock = "select stock from producto where id_producto=$cod_product limit 1 ";
    $rs_check_stock = mysqli_query($con, $sql_check_stock);
    $stock_re = mysqli_fetch_array($rs_check_stock);
    $stock = $stock_re['stock'];
}
```

Figura 43. Código salida

Elaborado por: Daniel Argoti, José Portilla

En la figura No. 43 se muestra cómo se procesa una salida dentro del sistema. Lo que se al momento de generar una salida, es buscar dentro la base de datos si la cantidad que se requiere está disponible dentro del stock. Si no es así no se puede generar la venta, caso contrario se genera la salida, y el valor de la cantidad de salida se resta del stock actual.

3.3 Pruebas de software

Pruebas de caja negra

Tabla 25. Detalle de pruebas de caja negra

Entrada	Salida	Resultado esperado
Acceso al sistema: ingresar usuario – contraseña y presionar iniciar sesión.	Página con menú principal	Si
Registro de productos: ingreso de datos de un producto y presionar en guardar.	Mensaje de notificación: “Datos guardados exitosamente”	Si
Consulta de clientes: presionar en la opción consultar clientes.	Listado de clientes	Si
Salida de productos: ingreso de datos de salida y presionar en vender.	Mensaje de notificación: “Venta generada exitosamente”	Si

Nota: se detallan las pruebas de caja negra.

En la tabla No. 25 se muestran las entradas de datos que fueron puestas a prueba como: el acceso al sistema, registro de productos, consulta de clientes y salida de productos. Mediante la salida que se obtuvo se verifico si el resultado fue el esperado o no.

Pruebas de caja blanca

Tabla 26. Detalle de pruebas de caja blanca

Acción a ejecutar	Correcto	Incorrecto
Verificar que el precio del producto sea mayor a 0.	Si	
Comprobar que el stock se incremente al ingresar una nueva entrada de producto.	Si	
Tratar de vender un producto que tengo su stock en 0.	Si	

Nota: se detallan las pruebas de caja blanca.

En la tabla No. 26 se muestran las acciones que fueron puestas a prueba como: verificar que el precio del producto sea mayor a 0, comprobar que el stock se incremente al ingresar una nueva entrada de producto y tratar de vender un producto que tengo su stock en 0. Se hizo un seguimiento al código.

Pruebas de integración

Tabla 27. Detalles de pruebas de integración

Acción a ejecutarse	Componentes	Correcto	Incorrecto
Cargar datos del cliente en la nota de venta al ingresar su cedula.	Módulo clientes Módulo productos	Si	
Mostrar información de un producto usando el código de barras.	Base de datos Módulo productos	Si	
Acceso a lista de proveedores cuando se ingresa un producto nuevo.	Módulo proveedores Modulo productos	Si	

Nota: se detallan las pruebas de integración a realizarse.

En la tabla No. 27 se muestran las acciones que fueron puestas a prueba como: cargar datos del cliente en la nota de venta al ingresar su cedula, mostrar información de un producto usando el código de barras y acceso a lista de proveedores cuando se ingresa un producto nuevo. Estas acciones requerían de la interacción entre 2 componentes para generar un resultado.

CONCLUSIONES

- La construcción del sistema informático para el manejo de inventarios de la distribuidora Mateo permitió que se realice el inventario de sus productos y a través del sistema tener actualizadas todas las existencias, de esta manera llegando a optimizar tiempo, dinero y manejo de personal.
- El uso de la metodología en cascada fue determinante en el proyecto, porque al tener una metodología de desarrollo como guía se logró disminuir el tiempo de desarrollo del sistema, además permitió responder rápidamente a cambios inesperados que surgieron en el proyecto
- El inventario manual realizado permitió recolectar toda la información necesaria sobre los activos de la distribuidora, lo que permitió asignar estos datos al sistema de manejo de inventarios. Esto sirvió para que el sistema pueda arrancar y entrar en funcionamiento.

RECOMENDACIONES

- Se recomienda dar mantenimiento al sistema cada cierto tiempo para garantizar su correcto funcionamiento y si es necesario adaptarlo a cambios externos que puedan surgir referente al manejo de inventarios de la distribuidora, esto se hará previo a un estudio realizado por el encargado del sistema informático.
- Se recomienda agregar un módulo de facturación que lleve toda la contabilidad de la distribuidora, esto permitirá conocer las pérdidas y ganancias automáticamente cada vez que se haga el inventario de productos. Además, será de gran utilidad para la declaración de impuestos.

LISTA DE REFERENCIAS

- Almsaeed, A. (2017). Obtenido de <https://adminlte.io/>
- Apache. (2017). Obtenido de <https://www.apachefriends.org/es/index.html>
- Blokehead, T. (2016). *Scrum*. Babelcube Inc.
- Cortes, R. (2006). *Introducción al análisis de sistemas*. Costa Rica: EUNED.
- Fernandez, V. (2006). *Desarrollo de sistemas de información*. Barcelona: UPC.
- GitHub. (2017). Obtenido de <http://getbootstrap.com/>
- Goytia, L. L., & Gutierrez, A. (2014). *Programación orientada a objetos*. Grupo editorial Patria.
- Kendall, K. y. (2007). *Análisis y diseño de sistema*. México DF: Pearson y Educación.
- Lamarca, M. (2013). *Hipertexto*. Obtenido de <http://www.hipertexto.info/documentos/oohdm.htm>
- Martel, A. (2014). *Gestión Práctica de Proyectos Con Scrum*. CreateSpace.
- Martel, A. (2016). *Gestión practica de proyectos*.
- Miguez, M., & Bastos, A. (2006). *Gestion de stocks*. Ideas propias.
- Miranda, J. (2005). *Gestión de proyectos*. Bogotá: MMEditores.
- Network, M. D. (2017). Obtenido de <https://developer.mozilla.org/es/docs/Web/CSS>
- Oracle. (2017). Obtenido de <https://www.mysql.com/>
- Oracle. (2017). *NetBeans*. Obtenido de <https://netbeans.org/>
- School, C. (2017). Obtenido de <https://www.javascript.com/>
- Sommerville, I. (2005). *Ingeniería del software*. Pearson Educación.

ANEXOS

Nota: Para revisar los anexos de este trabajo diríjase el CD.